



Guía del usuario

Amazon CodeCatalyst



Amazon CodeCatalyst: Guía del usuario

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

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

Table of Contents

¿Qué es Amazon CodeCatalyst?	1
¿Con qué puedo hacer? CodeCatalyst	1
¿Cómo puedo empezar CodeCatalyst?	2
Obtén más información sobre CodeCatalyst	2
Conceptos	3
AWS Los espacios de ID del constructor están en CodeCatalyst	4
Espacios que admiten la federación de identidades en CodeCatalyst	4
Proyectos	4
Proyectos	4
Conexiones de cuentas	5
Conexiones de VPC	5
AWS ID de constructor	5
Perfiles de usuario en CodeCatalyst	6
Repositorios de origen	6
Confirmaciones	7
Entornos de desarrollo	7
Flujos de trabajo	8
Acciones	8
Problemas	8
Tokens de acceso personal (PAT)	9
Roles	9
Configuración	10
Registrarse y crear recursos	12
Creando tu primer espacio y tus funciones de IAM	13
Acepta una invitación	19
Aceptar una invitación y crear un Builder ID AWS	20
Inicia sesión con tu ID de AWS constructor	21
Dispositivos de confianza	22
Aceptar una invitación por correo electrónico para iniciar sesión con SSO	22
Iniciar sesión con SSO	23
Ver todos los espacios y proyectos de un usuario	23
Visualización y administración de CodeCatalyst perfiles	24
Ver tu CodeCatalyst perfil	25
Ver el CodeCatalyst perfil de otro usuario	25

Actualización de tu perfil	26
Cambiar la CodeCatalyst contraseña	27
Configuración para usar el AWS CLI con CodeCatalyst	27
Tutoriales de introducción	30
Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles	31
Requisitos previos	33
Paso 1: Cree un proyecto de aplicación web moderno de tres niveles	34
Paso 2: Invita a alguien a tu proyecto	35
Paso 3: Crea temas para colaborar y realizar un seguimiento del trabajo	36
Paso 4: Visualiza tu repositorio de código fuente	37
Paso 5: Crea un entorno de desarrollo con una rama de prueba y realiza un cambio de código rápido	38
Paso 6: Visualiza el flujo de trabajo que crea la aplicación moderna	40
Paso 7: Pide a otras personas que revisen tus cambios	43
Paso 8: cerrar el problema	47
Eliminar recursos	47
Referencia	48
Tutorial: Empezar con un proyecto vacío	50
Requisitos previos	51
Cree un proyecto vacío	51
Crea un repositorio de origen	51
Crea un flujo de trabajo para crear, probar e implementar un cambio de código	53
Invita a alguien a tu proyecto	53
Crea temas en los que colaborar y hacer un seguimiento del trabajo	54
Tutorial: Uso de funciones de IA generativa	55
Requisitos previos	56
Agrega un resumen generado automáticamente al crear una solicitud de extracción	57
Crea un resumen de los comentarios dejados sobre los cambios de código en una solicitud de extracción	60
Crea una incidencia y asígnala a Amazon Q	61
Eliminar recursos	67
Tutorial: Cómo crear una aplicación completa con planos de PDK componibles	67
Requisitos previos	69
Paso 1: Crear un proyecto de monorepo	70
Paso 2: Agrega la API Type Safe al proyecto	71
Paso 3: Añadir un sitio web de Cloudscape React para el proyecto	73

Paso 4: generar la infraestructura para implementar la aplicación en la nube de AWS	74
Paso 5: Configure un DevOps flujo de trabajo para implementar su proyecto	76
Paso 6: Confirma el flujo de trabajo de publicación y consulta tu sitio web	77
Colabora e itera en el proyecto PDK	83
Espacios	100
Crear un espacio que admita a los usuarios de AWS Builder ID	102
Edición de un espacio	105
Eliminar un espacio	105
Supervisión de la actividad de un espacio	107
Cuentas de AWS Administrar un espacio	107
Añadir un elemento Cuenta de AWS a un espacio	108
Añadir funciones de IAM a las conexiones de cuentas	112
Añadir las funciones de IAM y de conexión de la cuenta a su entorno de implementación ...	113
Ver las conexiones de la cuenta	115
Eliminar una cuenta de un espacio (en CodeCatalyst)	115
Gestión de las funciones de IAM para las cuentas conectadas	116
Rol de CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i>	117
Rol de AWSRoleForCodeCatalystSupport	119
Crear un rol de IAM y usar la política de CodeCatalyst confianza	120
Gestión de los usuarios del espacio	121
Ver los miembros de un espacio	121
Invitar a un usuario directamente a un espacio	123
Cancelar una invitación a un espacio	124
Cambiar el rol de un miembro del espacio	125
Eliminar un elemento del espacio	125
Eliminar o cambiar el rol de un usuario con el rol de administrador de Space	126
Gestión de equipos	128
Crear un equipo	128
Ver un equipo	131
Administrar el rol de espacio de un equipo	131
Administrar un rol de proyecto para un equipo	132
Añadir un usuario a un equipo directamente	133
Eliminar directamente a un usuario de un equipo	134
Añadir un grupo de SSO a un equipo	134
Eliminar un equipo	135
Administrar los recursos de la máquina	136

Ver los recursos de la máquina	136
Desactive los recursos de la máquina	136
Habilite los recursos de la máquina	137
Administración de entornos de desarrollo para un espacio	138
Visualización de los entornos de desarrollo de su espacio	139
Edición de un entorno de desarrollo para tu espacio	140
Detener un entorno de desarrollo para su espacio	140
Eliminar un entorno de desarrollo para tu espacio	141
Cuotas de espacios	141
Proyectos	143
Creación de un proyecto	144
Crear un proyecto con un plano	145
Crear un proyecto vacío	145
Crear un proyecto con un GitHub repositorio vinculado	146
Visualización de un proyecto	149
Visualización de las tareas del proyecto y los entornos de desarrollo	149
Ver todos los proyectos	150
.....	150
Visualización de la configuración del proyecto	150
Cambiar a un proyecto diferente en CodeCatalyst	151
Eliminación de un proyecto	151
Administrar los miembros del proyecto	152
Ver los miembros de un proyecto	152
Invitar a un usuario a tu proyecto	154
Cancelar una invitación	155
Eliminar un usuario de tu proyecto	155
Aceptar o rechazar una invitación para un proyecto	156
Gestión de equipos para proyectos	156
Añadir un equipo a un proyecto	157
Administrar un rol de proyecto para un equipo	157
Eliminar un rol de proyecto para un equipo	158
Administrar los recursos de la máquina	158
Ver los recursos de la máquina	159
Desactive los recursos de la máquina	159
Habilite los recursos de la máquina	160
Cuotas para proyectos	161

Uso de notificaciones	162
¿Cómo funcionan las notificaciones?	163
Cómo empezar con las notificaciones de Slack	164
Administrar las notificaciones	168
Proyectos	174
Crear un proyecto con un plano	175
Aplicación y disociación de planos en un proyecto	175
Aplicar un plano a un proyecto	176
Desasociar un plano de un proyecto	177
Actualización de un plano en un proyecto	177
Edición de la descripción de un plano de un proyecto	179
Trabajar con la administración del ciclo de vida como usuario de blueprint	179
Utilizar la gestión del ciclo de vida en los proyectos existentes	180
Utilizar la gestión del ciclo de vida en varios planos de un proyecto	180
Trabajar con conflictos en el ciclo de vida de las solicitudes de cambios	180
Optar por no participar en los cambios en la administración del ciclo de vida	180
Anular la gestión del ciclo de vida de un plan en un proyecto	181
Referencia del plano del proyecto	181
Planos disponibles	182
Búsqueda de información sobre el plano del proyecto	187
Trabajar con planos personalizados	187
Conceptos de planos personalizados	188
Cómo empezar con los planos personalizados	192
Tutorial: Creación y actualización de una aplicación de React	196
Trabajar con la administración del ciclo de vida como autor de un plan	204
Desarrollo de un plan personalizado	210
Publicar un blueprint personalizado	241
Visualización de los detalles, las versiones y los proyectos de un blueprint personalizado ...	246
Añadir y eliminar un plano personalizado en un espacio	247
Administrar los permisos de publicación de un blueprint personalizado	248
Administración de versiones para un blueprint personalizado	249
Eliminar un blueprint o una versión personalizados publicados	250
Trabajar con dependencias y herramientas	252
Contribuye	254
Cuotas para los planos	255
Repositorios de fuentes	256

Conceptos de repositorio de fuentes	258
Proyectos	4
Repositorios de origen	258
Entornos de desarrollo	7
Tokens de acceso personal (PAT)	9
Ramificaciones	260
Ramas predeterminadas	260
Confirmaciones	7
Solicitudes de extracción	261
Revisiones	261
Flujos de trabajo	8
Configuración	262
Instalar Git	263
Crea un token de acceso personal	263
Cómo empezar con los repositorios de código fuente	264
Crear un proyecto mediante un plano	264
Ver los repositorios de un proyecto	266
Creación de un entorno de desarrollo	267
Creación de una solicitud de extracción	269
Fusionar una solicitud de extracción	271
Visualización del código desplegado	272
Limpieza de recursos	273
Trabajar con repositorios de código fuente	273
Crear un repositorio de código fuente	275
Vincular un repositorio de fuentes	276
Visualización de un repositorio de fuentes	277
Edición de la configuración de un repositorio de fuentes	278
Clonación de un repositorio fuente	279
Eliminar un repositorio de fuentes	281
Uso de las ramificaciones	283
Crear y eliminar una rama	284
Ver y cambiar la rama predeterminada de un repositorio	285
Administra las reglas de la sucursal	287
Comandos de Git para sucursales	289
Visualización de sucursales y detalles	291
Trabajar con archivos	292

Crear o añadir un archivo	292
Visualización de un archivo	295
Edición de un archivo	296
Cambiar el nombre o eliminar un archivo	297
Trabajo con solicitudes de extracción	297
Creación de una solicitud de extracción	299
Visualización de solicitudes de cambios	303
Gestiona las reglas de aprobación	306
Revisión de una solicitud de extracción	307
Actualización de una solicitud de extracción	311
Fusión de una solicitud de extracción	313
Cerrar una solicitud de extracción	317
Uso de confirmaciones	317
Visualización de las confirmaciones a una sucursal	318
Cambiar la forma en que se muestran las confirmaciones (CodeCatalystconsola)	319
Cuotas para los repositorios de fuentes	320
Entornos de desarrollo	325
Creación de un entorno de desarrollo	326
Entornos de desarrollo integrados compatibles para entornos de desarrollo	327
Crear un entorno de desarrollo en CodeCatalyst	327
Crear un entorno de desarrollo en un IDE	330
Detención de un entorno de desarrollo	331
Reanudación de un entorno de desarrollo	332
Edición de un entorno de desarrollo	334
Eliminación de un entorno de desarrollo	334
Conexión a un entorno de desarrollo mediante SSH	335
Configuración de su entorno de desarrollo	337
Edición de un archivo de desarrollo de repositorio para un entorno de desarrollo en CodeCatalyst	339
Edición de un archivo de desarrollo del repositorio para un entorno de desarrollo en un IDE	340
Mover un archivo de desarrollo del repositorio a un entorno de desarrollo	340
Modo de recuperación	340
Funciones de Devfile compatibles con CodeCatalyst	341
Ejemplo: configurar el archivo de desarrollo para su entorno de desarrollo	341
Comandos de Devfile	342

Eventos de Devfile	344
Componentes de Devfile	344
Imágenes universales devfile	345
Uso de entornos de desarrollo con una conexión de VPC	351
Uso de entornos de desarrollo con un IDE	352
Cuotas para entornos de desarrollo	352
Paquetes	354
Conceptos de paquetes	355
Paquetes	355
Espacios de nombres de paquetes	355
Versiones de paquetes	355
Activos	356
Repositorios de paquetes	356
Los repositorios Gateway	356
Repositorios upstream	357
Trabajar con repositorios de paquetes	357
Crear un repositorio de paquetes	357
Conectarse a un repositorio de paquetes	358
Edición de un repositorio de paquetes	358
Eliminar un repositorio de paquetes	359
Trabajar con repositorios ascendentes	359
Añadir un repositorio ascendente	360
Edición del orden de búsqueda de los repositorios ascendentes	361
Solicitar una versión de paquete con repositorios ascendentes	362
Eliminar un repositorio anterior	365
Conexión a repositorios públicos externos	366
Repositorios de paquetes externos compatibles y sus repositorios de puerta de enlace	367
Trabajar con paquetes	367
Publicación de paquetes	367
Ver los detalles de la versión del paquete	369
Eliminar una versión de paquete	369
Actualización del estado de la versión de un paquete	369
Edición de los controles de origen del paquete	371
Con npm	377
Configuración y uso de npm	377
control de etiquetas npm	387

Cuotas para paquetes	388
Cree, pruebe e implemente con flujos de trabajo	390
Acerca del archivo de definición del flujo de trabajo	390
Uso de los CodeCatalyst editores visuales y YAML de la consola	392
Descubriendo flujos de trabajo	394
Ver los detalles de ejecución del flujo de trabajo	395
Siguiendo pasos	396
Conceptos de flujos de	396
Flujos de trabajo	396
Archivos de definición del flujo de trabajo	397
Acciones	397
Grupos de acciones	397
Artefactos	398
Cálculo	398
Entornos	398
Informes	398
Ejecuciones	399
Orígenes	399
Variables	399
Activadores del flujo de	399
Cómo empezar con los flujos de trabajo	400
Requisitos previos	400
Paso 1: Crea y configura tu flujo de trabajo	401
Paso 2: Guarda tu flujo de trabajo con una confirmación	403
Paso 3: Ver los resultados de la ejecución	403
(Opcional) Paso 4: limpiar	404
Visualización de la calidad del código y el estado de la implementación por confirmación	404
Construir mediante flujos de trabajo	406
¿Cómo puedo crear una aplicación?	406
Ventajas de la acción de creación	407
Alternativas a la acción de construcción	408
Añadir la acción de construcción	408
Visualización de los resultados	410
Tutorial: Carga artefactos a Amazon S3	410
Pruebas mediante flujos de trabajo	419
Tipos de informes de prueba	420

Añadir la acción de prueba	423
Configuración de informes	423
Uso universal-test-runner	439
Prácticas recomendadas	441
Trabajando con pruebas	445
Implementación mediante flujos de trabajo	447
¿Cómo puedo implementar una aplicación?	448
Lista de acciones de despliegue	449
Ventajas de las acciones de despliegue	449
Alternativas para implementar acciones	450
Tutorial: Implemente mediante CloudFormation	451
Tutorial: Implementación en Amazon ECS	478
Tutorial: Implementación en Amazon EKS	514
Añadir la acción «Desplegar AWS CloudFormation pila»	546
Añadir la acción «Implementar en Amazon ECS»	550
Añadir la acción «Implementar en un clúster de Kubernetes»	553
Añadir la acción «AWS CDK desplegar»	557
Trabajo con implementaciones	564
Trabajar con flujos de trabajo	576
Crear, editar y eliminar un flujo de trabajo	577
Visualización del estado del flujo de trabajo	581
Trabajar con acciones	583
Trabajar con artefactos	646
Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución	660
Trabajo con entornos	685
Trabajar con el almacenamiento en caché de archivos	693
Trabajar con paquetes	697
Trabajar con corridas	702
Trabajando con secretos	715
Trabajar con fuentes	720
Trabajar con los factores desencadenantes	724
Trabajar con variables	742
Referencia de definición de flujo de trabajo	768
Ejemplo de un archivo de definición de flujo de trabajo	769
Pautas y convenciones de sintaxis	770
Propiedades de primer nivel	772

Referencia de acciones de compilación y prueba	785
Referencia de acción «Amazon S3 publish»	815
referencia a la AWS CDK acción «bootstrap»	825
referencia a la acción «AWS CDK desplegar»	837
«AWS Lambda invocar» la referencia a la acción	854
Referencia de acción «Desplegar AWS CloudFormation pila»	868
Referencia de acción «Implementar en Amazon ECS»	888
Referencia de acción «Implementar en un clúster de Kubernetes»	900
Referencia de acción «GitHub Acciones»	911
Referencia a la acción «Renderizar la definición de la tarea»	931
Cuotas para flujos de trabajo	940
Problemas	943
Conceptos de problemas	944
Problemas activos	944
Problemas archivados	945
Cesionario	945
Campos personalizados	945
Estimación	945
Problema	945
Etiqueta	946
Priority (Prioridad)	946
Estado y categorías de estado	946
Tareas	946
Vistas	946
Crear una incidencia	947
Prácticas recomendadas para los problemas asignados a Amazon Q	949
Edición y colaboración en temas	951
Edición de una edición	952
Uso de archivos adjuntos	953
Administrar las tareas relacionadas con los problemas	955
Marcar un problema como bloqueado o desbloqueado	955
Añadir, editar o eliminar comentarios	956
¿Está progresando con un problema	958
Archivar una incidencia	960
Problemas de búsqueda y visualización	961
¿Está buscando un problema	961

Problemas de clasificación	962
Problemas de agrupamiento	963
Problemas de filtrado	964
Crear una vista de problemas	964
Problemas de exportación	965
Configuración de los ajustes de problemas	965
Habilitar o deshabilitar varios cesionarios	966
Configurar la estimación del esfuerzo del problema	966
Estados	967
Etiquetas	969
Campos personalizados	970
Visualización y administración de los archivos adjuntos	971
Cuotas de emisiones	972
Identidad, permisos y acceso	974
Uso de roles de	975
Tipos de roles	975
Permisos disponibles para cada función	978
Ver y cambiar las funciones de los usuarios	1024
Administrar los tokens de acceso personal	1026
Creación de PAT	1026
Visualización de las PAT	1028
Eliminar las PAT	1030
Autenticación multifactorial (MFA) en Amazon CodeCatalyst	1031
¿Cómo registrar un dispositivo para su uso con la autenticación multifactorial	1031
Aplicaciones de autenticación	1034
Cómo cambiar los dispositivos MFA	1034
Seguridad	1036
Protección de datos	1037
CodeCatalyst e Identity and Access Management	1039
Validación de conformidad	1105
Resiliencia	1106
Seguridad de la infraestructura	1106
Configuración y análisis de vulnerabilidades	1107
Tus datos y privacidad en Amazon CodeCatalyst	1107
Mejores prácticas para las acciones de flujo de trabajo	1107
CodeCatalyst modelo de confianza	1109

Monitorización en Amazon CodeCatalyst	1111
Registrar las llamadas a la CodeCatalyst API a las cuentas conectadas en AWS	
CloudTrail	1114
Acceder a los eventos registrados en CodeCatalyst	1122
Cuotas de identidad, permisos y acceso	1125
Solución de problemas	1127
¿Problemas al registrarte	1127
¿Problemas al iniciar sesión	1128
¿Problemas para cerrar sesión	1129
Aparece el error «No existe un rol» debido a un flujo de trabajo fallido	1129
Aparece un error de rol debido a un flujo de trabajo fallido	1129
Necesito actualizar la función de IAM en el flujo de trabajo de un proyecto	1130
¿Cómo relleno un formulario de asistencia?	1130
Extensiones	1131
Conceptos de extensiones	1131
Extensiones	1131
CodeCatalyst catálogo	1131
Instalación y desinstalación de extensiones	1131
Instalación de una extensión	1132
Desinstalar una extensión	1132
Uso de GitHub repositorios en CodeCatalyst	1133
Inicio rápido: Uso de GitHub repositorios en CodeCatalyst	1134
Administrar GitHub cuentas	1138
Administrar repositorios GitHub	1140
Visualización de repositorios enlazados GitHub	1145
Uso de GitHub repositorios enlazados en los flujos de trabajo	1146
Uso de restricciones de direcciones IP en GitHub Enterprise Cloud	1148
El bloqueo de las solicitudes de GitHub extracción se fusiona cuando los flujos de trabajo fallan	1149
El uso de Jira tiene problemas en CodeCatalyst	1149
Inicio rápido: problemas con el uso de Jira en CodeCatalyst	1150
Administrar sitios de Jira	1154
Administrar proyectos de Jira	1157
Vincular las incidencias de Jira a las solicitudes de extracción	1160
Visualización de CodeCatalyst eventos en Jira	1161
Buscando problemas de Jira en CodeCatalyst	1162

Búsqueda	1163
Refinar tu consulta de búsqueda	1164
Refinar por tipo	1164
Refinar por campo	1165
Refinar con operadores booleanos	1165
Refinando por proyecto	1166
Consideraciones a la hora de trabajar con la búsqueda	1166
Referencia de campos en los que se pueden buscar	1167
Solución de problemas	1174
Solución de problemas generales de acceso	1174
He olvidado mi contraseña	1174
Parte o la totalidad de Amazon CodeCatalyst no está disponible	1175
No puedo crear un proyecto en CodeCatalyst	1175
Solución de problemas de soporte	1175
Aparece un error cuando accedo AWS Support a Amazon CodeCatalyst	1175
No puedo crear casos de soporte técnico para mi espacio	1176
Mi cuenta para casos de asistencia ya no está conectada a mi espacio en CodeCatalyst ..	1176
No puedo abrir un caso de soporte para otro Servicio de AWS en AWS Support Amazon CodeCatalyst	1177
Parte o la totalidad de Amazon CodeCatalyst no está disponible	1175
No puedo crear un proyecto en CodeCatalyst	1175
Quiero enviar comentarios en CodeCatalyst	1175
Solución de problemas de los repositorios de origen	1178
He alcanzado el límite máximo de almacenamiento de mi espacio y veo advertencias o errores	1179
Recibo un error al intentar clonar o enviar archivos a un repositorio de CodeCatalyst código fuente de Amazon	1179
Recibo un error al intentar confirmar o enviar información a un repositorio de CodeCatalyst código fuente de Amazon	1180
Necesito un repositorio de código fuente para mi proyecto	1180
Mi repositorio de código fuente es completamente nuevo, pero contiene una confirmación	1181
Quiero una rama diferente como mi rama predeterminada	1181
Estoy recibiendo correos electrónicos sobre la actividad de las solicitudes de incorporación de cambios	1181
He olvidado mi token de acceso personal (PAT)	1182
Una solicitud de cambios no muestra los cambios esperados	1182

Una solicitud de extracción muestra el estado de No se puede fusionar	1182
Solución de problemas de proyectos y planos	1183
A la API de Java le faltan dependencias en el AWS Fargate blueprint para apache-maven-3.8.6	1183
El flujo de trabajo moderno del blueprint de aplicaciones web de tres niveles	
OnPullRequestfalla debido a un error de permisos para Amazon CodeGuru	1184
¿Sigues intentando resolver tu problema?	1188
Resolución de flujos de trabajo	1188
¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?	1189
¿Cómo puedo corregir los errores «La definición de flujo de trabajo <i>no</i> tiene errores»?	1190
¿Cómo puedo corregir los errores «No se pueden localizar las credenciales» y ExpiredToken «»?	1192
¿Cómo puedo corregir los errores «No se puede conectar al servidor»?	1194
¿Por qué faltan CodeDeploy campos en el editor visual?	1194
¿Cómo puedo corregir los errores de las capacidades de IAM?	1195
¿Cómo puedo corregir los errores de «npm install»?	1197
¿Por qué varios flujos de trabajo tienen el mismo nombre?	1201
¿Puedo almacenar mis archivos de definición de flujo de trabajo en otra carpeta?	1201
¿Cómo puedo añadir acciones secuencialmente a mi flujo de trabajo?	1201
¿Por qué mi flujo de trabajo se valida correctamente pero falla en tiempo de ejecución? ...	1202
La detección automática no detecta ningún informe sobre mi acción	1202
Mi acción falla en los informes detectados automáticamente después de configurar los criterios de éxito	1203
La detección automática genera informes que no deseo	1203
La detección automática genera muchos informes pequeños para un único marco de pruebas	1204
Los flujos de trabajo enumerados en CI/CD no coinciden con los del repositorio de origen	1204
No puedo crear ni actualizar flujos de trabajo	1205
Solución de problemas de búsqueda	1206
No encuentro un usuario en mi proyecto	1206
No veo lo que busco en mi proyecto o espacio	1206
El número de resultados de búsqueda cambia constantemente cuando navego por las páginas	1207
Mi consulta de búsqueda no se está completando	1207
Solución de problemas con las cuentas asociadas	1207
Mi solicitud de Cuenta de AWS conexión recibe un error de token no válido	1208

CodeCatalyst El flujo de trabajo de mi proyecto de Amazon falla debido a un error en la cuenta, el entorno o el rol de IAM configurados	1208
Necesito una cuenta, un rol y un entorno asociados para crear un proyecto	1209
No puedo acceder a la página de Amazon CodeCatalyst Spaces en el AWS Management Console	1210
Quiero una cuenta diferente a la de facturación	1181
Solución de problemas de entornos de desarrollo	1211
La creación de mi entorno de desarrollo no se ha realizado correctamente debido a un problema con las cuotas	1212
No puedo enviar cambios de mi entorno de desarrollo a una rama específica de un repositorio	1212
Mi entorno de desarrollo no se reanudó	1213
Mi entorno de desarrollo se ha desconectado	1213
Falló mi entorno de desarrollo conectado a VPC	1213
No encuentro el directorio en el que se encuentra mi proyecto	1213
No puedo conectarme a mi entorno de desarrollo a través de SSH	1214
No puedo conectarme a mi entorno de desarrollo a través de SSH porque falta mi configuración de SSH local	1214
No puedo conectarme a mi entorno de desarrollo a través de SSH porque tengo problemas con mi perfil AWS Configcodecatalyst	1214
Solución de problemas de IDE	1214
Solución de problemas de devfiles	1216
Solución de problemas con	1219
No puedo elegir un responsable para mi problema	1219
Solución de problemas AWS CLI y problemas con el SDK	1219
Recibo un mensaje de error cuando introduzco en la línea de comandos o aws codecatalyst en el terminal el mensaje de error que indica que la elección no es válida	1219
Recibo un error de credenciales al ejecutar aws codecatalyst comandos	1220
CodeCatalyst informe de salud	1221
CodeCatalyst conceptos de informes de salud	1221
Incidente	1222
Estado	1222
Capacidades afectadas	1222
Actualizado el	1222
AWS Supportpara Amazon CodeCatalyst	1223
Facturación AWS Support para Amazon CodeCatalyst	1223

Configurar tu espacio AWS Support para Amazon CodeCatalyst	1226
Acceder al soporte CodeCatalyst en el AWS Management Console	1227
Crear un caso CodeCatalyst de soporte en CodeCatalyst	1228
Resolver un caso de soporte en CodeCatalyst	1231
Reabrir un caso de soporte en CodeCatalyst	1231
Cuotas	1233
Historial de documentos	1235
AWS Glosario	1259
.....	mclx

¿Qué es Amazon CodeCatalyst?

Amazon CodeCatalyst es un servicio integrado para los equipos de desarrollo de software que adoptan prácticas de integración e implementación continuas en sus procesos de desarrollo de software. CodeCatalyst pone todas las herramientas que necesita en un solo lugar. Puede planificar el trabajo, colaborar en el código y crear, probar e implementar aplicaciones con herramientas de integración continua y entrega continua (CI/CD). También puede integrar AWS los recursos con sus proyectos conectándolos a su Cuentas de AWS espacio. CodeCatalyst Al gestionar todas las etapas y aspectos del ciclo de vida de las aplicaciones en una sola herramienta, puede entregar software de forma rápida y segura.

En CodeCatalyst, crea un espacio para representar a su empresa, departamento o grupo y, a continuación, crea proyectos que contienen los recursos necesarios para respaldar sus equipos y tareas de desarrollo. CodeCatalyst los recursos se estructuran dentro de proyectos que viven dentro de espacios. Para ayudar a los equipos a empezar rápidamente, CodeCatalyst proporciona planos de proyectos basados en el lenguaje o en las herramientas. Cuando creas un proyecto a partir de un esquema de proyecto, el proyecto incluye recursos como un repositorio de código fuente con código de muestra, scripts de compilación, acciones de implementación, servidores virtuales o recursos sin servidor, y más.

¿Con qué puedo hacer? CodeCatalyst

Usted y su equipo de desarrollo pueden utilizarlos CodeCatalyst para llevar a cabo todos los aspectos del desarrollo de software, desde la planificación del trabajo hasta la implementación de las aplicaciones. Puede utilizar CodeCatalyst para:

- Realice iteraciones y colabore en el código: trabaje en colaboración con su equipo en el desarrollo del código mediante ramificaciones, combinaciones, solicitudes de extracción y comentarios en sus repositorios de código fuente. Crea entornos de desarrollo para trabajar en el código rápidamente sin tener que clonar ni configurar conexiones a los repositorios.
- Cree, pruebe e implemente su aplicación con flujos de trabajo: configure los flujos de trabajo con acciones de creación, prueba e implementación para gestionar la integración y la entrega continuas de sus aplicaciones. Puedes iniciar los flujos de trabajo manualmente o configurarlos para que se inicien automáticamente en función de eventos como la inserción de códigos o la creación o el cierre de solicitudes de cambios.

- Da prioridad al trabajo de tu equipo mediante el seguimiento de las incidencias: utiliza las incidencias para crear acumulaciones y monitorizar el estado de las tareas en curso mediante tableros. Crear y mantener una buena cantidad de elementos pendientes para que tu equipo pueda trabajar es una parte importante del desarrollo de software.
- Configura la supervisión y las notificaciones: supervisa la actividad del equipo y el estado de los recursos, y configura las notificaciones para estar al día de los cambios importantes.

¿Cómo puedo empezar CodeCatalyst?

Si no tienes un espacio o quieres aprender a configurar y administrar un espacio, te recomendamos que comiences con la [Guía del CodeCatalyst administrador de Amazon](#).

Si es la primera vez que trabajas en un proyecto o un espacio, te recomendamos que comiences de la siguiente manera:

- Revisando el [CodeCatalyst conceptos](#)
- [Crear un espacio que admita a los usuarios de AWS Builder ID](#)
- Cree su primer proyecto siguiendo los pasos de [Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles](#)

Obtén más información sobre CodeCatalyst

Puede obtener más información sobre las funciones de CodeCatalyst en esta guía del usuario, así como en los siguientes recursos:

- [AWS DevOps Artículos de blog sobre Amazon CodeCatalyst](#)
- [La guía de referencia CodeCatalyst de las API de Amazon](#)
- [Guía para desarrolladores del Amazon CodeCatalyst Action Development Kit](#)
- [CodeCatalyst Preguntas frecuentes](#)
- [Testimonios](#)

CodeCatalyst conceptos

Familiarízate con los conceptos clave que te ayudarán a acelerar la colaboración y el desarrollo de aplicaciones en Amazon CodeCatalyst. Estos conceptos incluyen términos utilizados en el control de código fuente, la integración y la entrega continuas (CI/CD) y el modelado y la configuración de los procesos de publicación automatizados.

Para obtener información conceptual adicional, consulte los siguientes temas:

- [Conceptos de repositorio de fuentes](#)
- [Conceptos de flujos de](#)

Temas

- [AWS Los espacios de ID del constructor están en CodeCatalyst](#)
- [Espacios que admiten la federación de identidades en CodeCatalyst](#)
- [Proyectos](#)
- [Proyectos](#)
- [Conexiones de cuentas](#)
- [Conexiones de VPC](#)
- [AWS ID de constructor](#)
- [Perfiles de usuario en CodeCatalyst](#)
- [Repositorios de origen](#)
- [Confirmaciones](#)
- [Entornos de desarrollo](#)
- [Flujos de trabajo](#)
- [Acciones](#)
- [Problemas](#)
- [Tokens de acceso personal \(PAT\)](#)
- [Roles](#)

AWS Los espacios de ID del constructor están en CodeCatalyst

El administrador del espacio invita a los usuarios a CodeCatalyst entrar enviando correos electrónicos de invitación individuales desde la página de miembros. Los usuarios que estén invitados o se registren para CodeCatalyst crear su propio AWS Builder ID. El perfil se administra en AWS Builder ID y se muestra como nombre de usuario e información de perfil en la configuración de usuario de CodeCatalyst.

Espacios que admiten la federación de identidades en CodeCatalyst

Los usuarios que se han añadido a los usuarios y grupos del SSO de la instancia del Centro de Identidad de IAM y que se administran en el almacén de identidades y son invitados a su espacio a través del Centro de Identidad de IAM. El administrador de Space sincroniza la página de CodeCatalyst miembros para ver las actualizaciones más recientes. Los usuarios inician sesión mediante el portal de inicio de sesión SSO configurado en la instancia del IAM Identity Center de la empresa. Los espacios que admiten la federación de identidades se conectan a la instancia del almacén de identidades a través de la aplicación Identity Center y su asignación al ID del almacén de identidades.

Proyectos

Un proyecto representa un esfuerzo de colaboración CodeCatalyst que apoya las tareas y los equipos de desarrollo. Una vez que tengas un proyecto, puedes añadir, actualizar o eliminar usuarios y recursos, personalizar el panel de control del proyecto y supervisar el progreso del trabajo de tu equipo. Puede tener varios proyectos dentro de un espacio.

Para obtener más información sobre los proyectos, consulte [Proyectos en CodeCatalyst](#).

Proyectos

Un blueprint es un sintetizador de proyectos que genera y amplía las dependencias y los archivos de soporte de las aplicaciones, además de crear el CodeCatalyst proyecto en la consola. Puede elegir un tipo de proyecto de una selección de planos CodeCatalyst, ver el archivo README y obtener una vista previa del repositorio del proyecto y los recursos que se generarán. El proyecto se genera a partir de la configuración básica especificada en el plano. El esquema del proyecto se sintetiza

periódicamente, lo que actualiza los archivos del proyecto, como las dependencias de software, y regenera los recursos. Los proyectos utilizan una herramienta llamada Projen para sintetizar los proyectos mediante la sincronización de las últimas actualizaciones del proyecto y la generación de archivos de soporte. Estos archivos pueden incluir `package.json`, y más `Makefile` `eslint`, según el tipo y el idioma de la aplicación. Los planos de proyectos pueden generar archivos que AWS respalden recursos como componentes, AWS CloudFormation plantillas y plantillas de CDK. AWS Serverless Application Model

Para obtener más información sobre los esquemas de proyectos, consulte [Referencia del plano del proyecto](#)

Conexiones de cuentas

Una conexión de cuenta asocia un CodeCatalyst espacio a su. Cuenta de AWS Una vez configurada la conexión de su cuenta, estará Cuenta de AWS disponible en el espacio. A continuación, puede añadir funciones de IAM para que CodeCatalyst pueda acceder a los recursos de su Cuenta de AWS cuenta. También puede utilizar estas funciones para las acciones de su CodeCatalyst flujo de trabajo.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#).

Conexiones de VPC

Una conexión de VPC es un CodeCatalyst recurso que contiene todas las configuraciones necesarias para que su flujo de trabajo acceda a una VPC. Los administradores del espacio pueden añadir sus propias conexiones de VPC en la CodeCatalyst consola de Amazon en nombre de los miembros del espacio. Al agregar una conexión de VPC, los miembros del espacio pueden ejecutar acciones de flujo de trabajo y crear entornos de desarrollo que cumplan con las reglas de la red y puedan acceder a los recursos de la VPC asociada.

Para obtener más información sobre las conexiones de VPC, consulte Gestión de [Amazon Virtual Private Clouds](#) en la Guía del CodeCatalyst administrador.

AWS ID de constructor

Un AWS Builder ID es una identidad personal que puedes usar para registrarte e iniciar sesión en ellas CodeCatalyst y en otras aplicaciones participantes. No es lo mismo que un Cuenta de AWS.

Su ID de AWS creador administra metadatos como el alias de usuario y la dirección de correo electrónico. Tu ID de AWS creador es una identidad única que apoya a los usuarios en todos los espacios de CodeCatalyst. Para obtener información sobre cómo acceder a su perfil de AWS Builder ID, consulte [Actualización de tu perfil](#). Para obtener más información sobre AWS Builder ID, consulte [AWS Builder ID](#) en Referencia general de AWS.

Para obtener más información sobre cómo registrarse e iniciar sesión, consulte [Configuración CodeCatalyst](#).

Perfiles de usuario en CodeCatalyst

Para acceder a su perfil de CodeCatalyst usuario, elija la opción de perfil en el menú desplegable situado debajo de sus iniciales de inicio de sesión en cualquier página de CodeCatalyst. Puedes crear identificadores de acceso personal (PAT) desde tu página de perfil, pero solo puedes ver o eliminar los PAT con ellos. AWS CLI Su nombre de usuario es el alias que eligió al registrarse. No puede cambiar su nombre de usuario. Para ver la página de perfil de otro CodeCatalyst usuario, vaya a la pestaña Miembros de su proyecto y elija el usuario adecuado.

Para acceder a su ID de AWS constructor, consulte su CodeCatalyst perfil y, a continuación, elija ir a AWS Builder ID. Se le redirigirá a su página de perfil de AWS Builder ID. El nombre completo, la dirección de correo electrónico y la contraseña de su perfil se administran mediante su ID de AWS Builder, y puede editar esa información en la página de AWS Builder ID. Introdujiste esta información cuando te registraste. Cuando esté listo para configurar MFA para usar una aplicación de autenticación para iniciar sesión, utilizará la página AWS Builder ID. Para obtener más información sobre cómo ver su perfil de AWS Builder ID, consulte [Actualización de tu perfil](#).

Para obtener más información sobre cómo registrarse e iniciar sesión, consulte [Configuración CodeCatalyst](#).

Repositorios de origen

Un repositorio de código fuente es el lugar donde se almacenan de forma segura el código y los archivos de un proyecto. También almacena el historial de versiones de los archivos. De forma predeterminada, el repositorio de origen se comparte con los demás usuarios CodeCatalyst del proyecto. Puedes tener más de un repositorio de código fuente para un proyecto. Puedes crear repositorios de código fuente para proyectos en CodeCatalyst él, o puedes elegir vincular un repositorio de código fuente existente hospedado por otro servicio si ese servicio es compatible con una extensión instalada. Por ejemplo, puedes vincular un GitHub repositorio a un proyecto después

de instalar la extensión GitHub Repositories. Para obtener más información, consulte [Trabajar con repositorios de código fuente en CodeCatalyst](#) y [Inicio rápido: Uso de GitHub repositorios en CodeCatalyst](#).

Los repositorios de origen también son donde se almacena la información de configuración CodeCatalyst del proyecto, como el archivo de configuración que define los atributos y las acciones del flujo de trabajo de CI/CD. Si crea el proyecto mediante un plano, se creará un repositorio de origen con la información de configuración del proyecto almacenada en su interior. Si crea un proyecto vacío, debe crear un repositorio de origen antes de poder crear recursos que requieran información de configuración, como los flujos de trabajo.

Para obtener más conceptos que pueden ayudarle a trabajar con los repositorios de código fuente y el control de código fuente, consulte [Conceptos de repositorio de fuentes](#).

Confirmaciones

Una confirmación es un cambio en un archivo o conjunto de archivos. En la CodeCatalyst consola de Amazon, una confirmación guarda los cambios y los envía a un repositorio de origen. La confirmación incluye información sobre el cambio, incluida la identidad del usuario que lo realizó, la fecha y hora del cambio, el título del compromiso y cualquier mensaje incluido sobre el cambio. Para obtener más información, consulte [Trabajar con compromisos en Amazon CodeCatalyst](#).

En el contexto de un repositorio fuente CodeCatalyst, las confirmaciones son instantáneas de los cambios en el contenido del repositorio. Cada vez que un usuario confirma e impulsa un cambio, CodeCatalyst guarda información que incluye quién realizó el cambio, la fecha y la hora de la confirmación y los cambios realizados como parte de la confirmación. También puedes añadir etiquetas Git a las confirmaciones para ayudar a identificar confirmaciones específicas.

Para obtener más información sobre las confirmaciones, consulta [Trabajar con compromisos en Amazon CodeCatalyst](#).

Entornos de desarrollo

Un entorno de desarrollo es un entorno de desarrollo basado en la nube que puede utilizar CodeCatalyst para trabajar rápidamente en el código almacenado en los repositorios de código fuente de su proyecto. Las herramientas del proyecto y las bibliotecas de aplicaciones incluidas en su entorno de desarrollo se definen mediante un archivo de desarrollo en el repositorio de código fuente de su proyecto. Si no tiene un archivo de desarrollo en su repositorio de código fuente, se

aplicará automáticamente un archivo de desarrollo predeterminado. El devfile predeterminado incluye herramientas para los marcos y lenguajes de programación más utilizados. De forma predeterminada, un entorno de desarrollo está configurado para tener un procesador de 2 núcleos, 4 GB de RAM y 16 GiB de almacenamiento persistente.

Flujos de trabajo

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Para obtener más información acerca de los flujos de trabajo, consulte [Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst](#).

Acciones

Una acción es el componente principal de un flujo de trabajo y define una unidad lógica de trabajo que se debe realizar durante la ejecución de un flujo de trabajo. Normalmente, un flujo de trabajo incluye varias acciones que se ejecutan de forma secuencial o paralela en función de cómo las haya configurado.

Para obtener más información sobre las acciones, consulte [Trabajar con acciones](#).

Problemas

Una edición es un registro que hace un seguimiento del trabajo relacionado con el proyecto. Puedes crear un problema para una función, una tarea, un error o cualquier otra obra relacionada con tu

proyecto. Si utilizas un desarrollo ágil, un problema también puede describir una historia épica o de usuario.

Para obtener más información sobre los problemas, consulte [Problemas en CodeCatalyst](#).

Tokens de acceso personal (PAT)

Un token de acceso personal (PAT) es similar a una contraseña. Está asociado a tu identidad de usuario para que lo utilices en todos los espacios y proyectos en los que se encuentre CodeCatalyst. Las PATs se utilizan para acceder a CodeCatalyst recursos que incluyen entornos de desarrollo integrados (IDE) y repositorios de código fuente basados en Git. Las PAT te representan CodeCatalyst y puedes gestionarlas en tu configuración de usuario. Un usuario puede tener más de una PAT. Los tokens de acceso personal solo se muestran una vez. Como práctica recomendada, asegúrese de guardarlos de forma segura en su ordenador local. De forma predeterminada, los PAT caducan al cabo de un año.

Para obtener más información sobre las PAT, consulte. [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#)

Roles

Un rol define el acceso de un usuario a los recursos de un proyecto o un espacio y las acciones que puede realizar ese usuario. Tú eliges el rol de un usuario cuando lo invitas a un proyecto. Hay roles a nivel de espacio y roles a nivel de proyecto. CodeCatalyst Un usuario con un rol administrativo en el nivel correcto puede cambiar los roles asignados. Por ejemplo, un usuario con el rol de administrador del proyecto tiene el control total sobre ese proyecto y puede cambiar los roles de los usuarios en ese proyecto. Para obtener información sobre los roles disponibles y los permisos que tiene cada rol, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Configuración CodeCatalyst

Hay dos tipos de espacios que se pueden configurar CodeCatalyst: los espacios que admiten a los usuarios de AWS Builder ID y la creación de un espacio que admite la federación de identidades, donde los usuarios y grupos de SSO se administran en el Centro de identidades de IAM. Los usuarios de un espacio de AWS Builder ID inician sesión CodeCatalyst con su ID de AWS Builder y los usuarios de un espacio empresarial inician sesión CodeCatalyst mediante el portal de SSO de la empresa asociada al espacio.

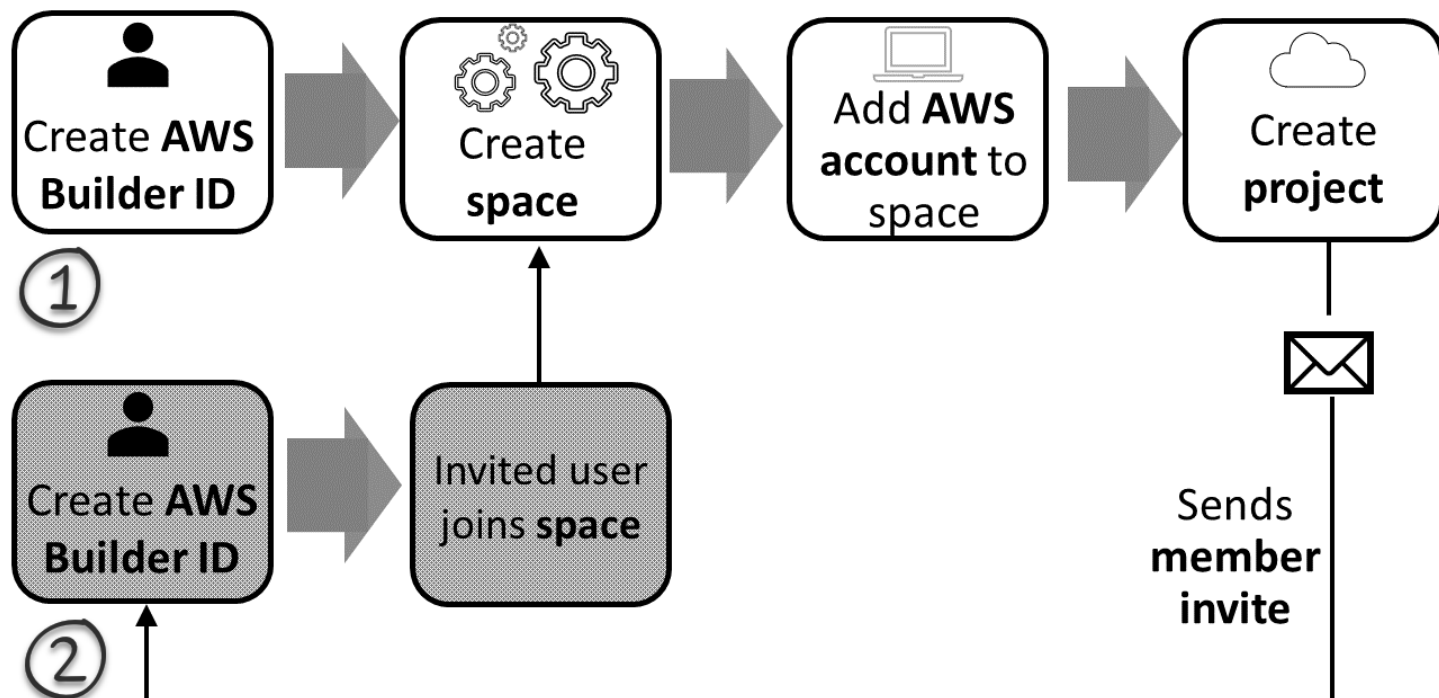
Los pasos para configurar y administrar un espacio AWS Builder ID se detallan en esta guía. Para trabajar con un espacio de CodeCatalyst AWS Builder ID, lo configurará CodeCatalyst utilizando la configuración de usuario y el AWS Builder ID que utilice para iniciar sesión CodeCatalyst.

Los pasos para configurar y administrar un espacio que admita la federación de identidades se indican en la Guía CodeCatalyst del administrador. Para trabajar con espacios configurados para la federación de identidades, consulte [Configuración y administración de CodeCatalyst espacios](#) en la Guía del CodeCatalyst administrador de Amazon.

Esta sección proporciona dos rutas comunes para configurar Amazon para trabajar CodeCatalyst con un espacio AWS Builder ID: crear un espacio y un proyecto como primer usuario y aceptar una invitación a un espacio o proyecto existente. Estos flujos de trabajo de configuración son necesariamente muy diferentes. El siguiente diagrama muestra ambos procesos de registro de la siguiente manera:

Existen dos formas comunes de prepararse para trabajar en Amazon CodeCatalyst: crear un espacio y un proyecto como primer usuario y aceptar una invitación a un espacio o proyecto existente. Estos flujos de trabajo de configuración son necesariamente muy diferentes. El siguiente diagrama muestra ambos procesos de registro de la siguiente manera:

1. En el primer caso, debe crear y configurar un espacio para su empresa, equipo o grupo y crear un proyecto antes de invitar a otras personas a utilizar estos recursos. Cuenta de AWS Debe proporcionarse una a efectos de facturación, ya que puede seguir utilizando la capa gratuita de forma predeterminada.
2. En el segundo caso, si te unes CodeCatalyst aceptando una invitación a un proyecto, otra persona ya ha creado un espacio y un proyecto para ti. Sin embargo, igual querrás configurar tu perfil para poder empezar a trabajar con otras personas.



i Tip

CodeCatalyst usa espacios para agrupar proyectos y recursos. Cuando te registres por primera vez CodeCatalyst, se te pedirá que crees un espacio y un proyecto.

Tanto si te registras para crear un espacio y un proyecto como si lo haces para aceptar una invitación, crearás un AWS Builder ID que utilizarás para iniciar sesión CodeCatalyst. Para crear un AWS Builder ID, debe proporcionar el nombre completo, la contraseña y la dirección de correo electrónico que utiliza para iniciar sesión en AWS las aplicaciones. CodeCatalyst Después de este punto, utiliza el correo electrónico y la contraseña para iniciar sesión. También puede usar este AWS Builder ID para iniciar sesión en otras aplicaciones que usen credenciales de AWS Builder ID.

En AWS Builder ID CodeCatalyst y en él, se genera un perfil en función de su información de inicio de sesión. Su perfil contiene sus CodeCatalyst preferencias de idioma y configuración de notificaciones en sus CodeCatalyst proyectos.

i Tip

Si tienes algún problema al registrarte en tu CodeCatalyst perfil de Amazon, sigue los pasos que se indican en esa página. Si necesitas más ayuda, consulta [¿Problemas al registrarte?](#)

Temas

- [Inscríbete para crear tu primer espacio y tu rol de desarrollo](#)
- [Aceptar una invitación y crear tu ID de AWS constructor](#)
- [Inicia sesión con tu ID de AWS constructor](#)
- [Aceptar una invitación por correo electrónico para iniciar sesión con SSO](#)
- [Iniciar sesión con SSO](#)
- [Ver todos los espacios y proyectos de un usuario](#)
- [Visualización y administración de CodeCatalyst perfiles](#)
- [Configuración para usar el AWS CLI con CodeCatalyst](#)

Inscríbete para crear tu primer espacio y tu rol de desarrollo

Puedes registrarte en Amazon CodeCatalyst sin una invitación a un espacio o proyecto existente. Cuando lo hagas, crearás un espacio y un proyecto después de crear tu ID de AWS constructor. Como parte de la creación de un espacio, tendrás que añadir una Cuenta de AWS para fines de facturación.

i Tip

Si tienes algún problema al registrarte en tu CodeCatalyst perfil de Amazon, sigue los pasos que se indican en esa página. Si necesitas más ayuda, consulta [¿Problemas al registrarte?](#)

Este es un flujo posible para un usuario que empieza CodeCatalyst sin una invitación a un proyecto o un espacio.

Mary Major es una desarrolladora que está interesada en CodeCatalyst y decide probarlo. Navega a la CodeCatalyst consola y elige la opción de registrarse y crear un AWS Builder ID. Mary proporciona

una dirección de correo electrónico y una contraseña para crear su ID de AWS constructor. Podrá usar su ID de AWS constructor para iniciar sesión en CodeCatalyst y en otras aplicaciones. Cuando se le pida que elija un alias, MaryMajor especificará el nombre de CodeCatalyst usuario que aparecerá CodeCatalyst y que otros miembros del proyecto utilizarán para @mention Mary.

A continuación, se le indica automáticamente a Mary que cree un espacio. Como parte de este flujo, se le pide a Mary que asocie una Cuenta de AWS al espacio que está creando para poder ver el código de muestra en la creación e implementación de su primer proyecto. Agrega esa información y crea su espacio, donde elige la opción de crear una función de desarrollo de vista previa que pueda usarse en los proyectos de su nuevo espacio. Mary elige crear un proyecto y, a continuación, ve una lista de planos de proyectos. Tras revisar la información de los planos disponibles, decide probar el modelo de aplicación web moderna de tres niveles para su primer proyecto. Rellena los campos obligatorios y crea el proyecto. Tan pronto como el proyecto esté listo, accederá a una página de resumen del proyecto que incluye la actividad reciente, así como enlaces al código del proyecto y al flujo de trabajo que crea e implementa automáticamente ese código. Explora tanto el código como el flujo de trabajo, incluida la visualización de la aplicación web de muestra desplegada. Al gustarle lo que ve, decide invitar a algunos de sus compañeros de trabajo al proyecto para empezar a CodeCatalyst explorarlo.

Cuando tiene un momento, Mary configura su AWS Builder ID para iniciar sesión CodeCatalyst con la autenticación multifactor (MFA). Con la MFA configurada, Mary puede iniciar sesión CodeCatalyst con una combinación de su CodeCatalyst contraseña y un código de acceso o token de una aplicación de autenticación de terceros aprobada.

Creando tu primer espacio y tus funciones de IAM

Sigue estos pasos para registrarte en tu CodeCatalyst perfil de Amazon, crear un espacio y añadir una cuenta, un rol de soporte y un rol de desarrollador para tu espacio.

El procedimiento final crea y añade el rol de desarrollador. El rol de desarrollador es un rol de AWS IAM que permite a sus CodeCatalyst flujos de trabajo acceder a AWS los recursos. El rol de desarrollador es un rol de servicio que se utiliza para administrar Servicios de AWS y se creará en la cuenta en la que se inicie sesión. Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. El rol tendrá un nombreCodeCatalystWorkflowDevelopmentRole-*spaceName*. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName*de servicio](#).

Note

Como práctica recomendada de seguridad, asigne el acceso administrativo únicamente a los usuarios administrativos y a los desarrolladores que necesiten gestionar el acceso a AWS los recursos del espacio.

Antes de empezar, debe estar preparado para proporcionar un Cuenta de AWS identificador para una cuenta en la que tenga privilegios administrativos. Tenga a mano su Cuenta de AWS ID de 12 dígitos. Para obtener información sobre cómo encontrar tu Cuenta de AWS ID, consulta [Tu Cuenta de AWS ID y su alias](#).

Para registrarse como usuario nuevo

1. Antes de empezar a usar la CodeCatalyst consola AWS Management Console, abre el espacio y asegúrate de que has iniciado sesión con el mismo Cuenta de AWS que deseas usar para crear tu espacio.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. En la página de bienvenida, elija Inscripción. Aparecerá la página Crear su ID de creador de AWS. Tu ID de AWS constructor es una identidad que creas para iniciar sesión. No es lo mismo que un Cuenta de AWS.
4. En Tu dirección de correo electrónico, ingresa la dirección de correo electrónico con la que deseas asociarte CodeCatalyst. A continuación, elija Siguiente.
5. En Tu nombre, introduce el nombre y apellidos que deseas que aparezcan en las aplicaciones en las que utilices tu ID de AWS constructor. Se permiten espacios. Este será tu nombre de perfil de AWS Builder ID, como Mary Major. Puedes cambiar el nombre más adelante.

Elija Siguiente. Aparece la página de verificación del correo electrónico.

6. Se enviará un código de verificación al correo electrónico que especificó. Introduce este código en Código de verificación y, a continuación, selecciona Verificar. Si no recibes el código después de 5 minutos y no lo encuentras en las carpetas de correo no deseado o basura, selecciona Reenviar el código.
7. Una vez que hayamos verificado tu código, introduce una contraseña que cumpla con los requisitos de Contraseña y Confirmar contraseña.


Selecciona la casilla de verificación para confirmar que estás de acuerdo con el acuerdo con el AWS cliente y las condiciones del AWS servicio y, a continuación, selecciona Crear ID de AWS constructor.

8. En la página Crea tu CodeCatalyst alias, introduce el alias que quieras usar como identificador de usuario único. CodeCatalyst Elija una versión abreviada de su nombre sin espacios, como MaryMajor. Otros CodeCatalyst usuarios la usarán para escribirte como @mention en sus comentarios y solicitudes de cambios. Tu CodeCatalyst perfil contendrá tu nombre completo de tu ID de AWS creador y tu CodeCatalyst alias. No puedes cambiar tu CodeCatalyst alias más adelante.

Tu nombre completo y tu alias aparecerán en diferentes áreas de CodeCatalyst. Por ejemplo, el nombre de tu perfil aparece para la actividad que aparece en la lista de actividades, pero los miembros del proyecto usarán tu alias como @mention you.

Elija Siguiente. La página se actualiza para mostrar la sección Crea tu CodeCatalyst espacio.

9. En Ponle un nombre a tu espacio, ingresa el nombre de tu espacio. No puedes cambiarlo más adelante.

 Note

Los nombres de los espacios deben ser únicos en todos ellos CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados.

10. En el menú Región de AWS desplegable, elige la región en la que deseas almacenar tu espacio y los datos del proyecto. No puedes cambiar esto más adelante.
11. Elija Siguiente. La página se actualiza para mostrar la página en la que se puede añadir unCuenta de AWS. Esta cuenta se utilizará como cuenta de facturación del espacio.
12. En Cuenta de AWSID, introduce el ID de doce dígitos de la cuenta que quieres conectar a tu espacio.

En el token de verificación de la AWS cuenta, copia el ID del token generado. El token se copia automáticamente, pero es posible que desee almacenarlo mientras aprueba la solicitud de AWS conexión.

13. Selecciona Ir a la AWS consola para verificarlo.


14. La página Verificar CodeCatalyst el espacio de Amazon se abre enAWS Management Console. Esta es la página de Amazon CodeCatalyst Spaces. Puede que tengas que iniciar sesión para acceder a la página.

En elAWS Management Console, asegúrate de elegir el mismo Región de AWS lugar en el que deseas crear tu espacio.

Para acceder directamente a la página, inicia sesión en Amazon CodeCatalyst Spaces AWS Management Console en <https://console.aws.amazon.com/codecatalyst/home/>.

El campo del token de verificación AWS Management Console se rellena automáticamente con el token generado en CodeCatalyst.

15. (Opcional) En los niveles de pago autorizados, selecciona Autorizar los niveles de pago (Estándar, Enterprise) para activar los niveles de pago en tu cuenta de facturación.

 Note

Esto no actualiza el nivel de facturación a un nivel de pago. Sin embargo, esto lo configura Cuenta de AWS para que puedas cambiar el nivel de facturación de tu espacio en cualquier momento. CodeCatalyst Puedes activar los niveles de pago en cualquier momento. Sin realizar este cambio, el espacio solo puede usar el nivel gratuito.

16. Seleccione Verificar espacio.

Aparece un mensaje de confirmación de la cuenta para indicar que la cuenta se ha agregado al espacio.

17. Permanece en la página Verify Amazon CodeCatalyst Space. Elija el siguiente enlace: Para añadir funciones de IAM a este espacio, consulte los detalles del espacio.


La página de conexiones con los detalles CodeCatalyst del espacio se abre en elAWS Management Console. Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

18. Vuelva a la CodeCatalyst página y, a continuación, seleccione Siguiente.

19. Aparece un mensaje de estado mientras se crea su espacio. Cuando se crea el espacio, aparece CodeCatalyst el siguiente mensaje: El espacio está listo. El último paso es crear un proyecto. . Puede elegir una de las opciones siguientes:

- Elige Omitir por ahora.

- Elige Crea tu primer proyecto para tu espacio. Para ver un tutorial que muestra cómo crear un proyecto con un plano, consulte [Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles](#)

 Note

Si aparece un mensaje o un error de permisos, actualice la página e intenta verla de nuevo.

Para crear y añadir el CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Antes de empezar en la CodeCatalyst consola, abra el espacio yAWS Management Console, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
4. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de AWSdetalles.
5. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM a Amazon CodeCatalyst Space se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

6. Elija Crear función CodeCatalyst de administrador de desarrollo en IAM. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza del rol de desarrollo. El rol tendrá un nombreCodeCatalystWorkflowDevelopmentRole-*spaceName*. Para obtener más información sobre el rol y la política de roles, consulte[Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName*de servicio](#).

 Note

Este rol solo se recomienda para su uso con cuentas de desarrollador y usa la política `AdministratorAccess` AWS administrada, lo que le da acceso total para crear nuevas políticas y recursos en ella Cuenta de AWS.

7. Selecciona Crear rol de desarrollo.
8. En la página de conexiones, en la sección Funciones de IAM disponibles para CodeCatalyst, consulta la `CodeCatalystWorkflowDevelopmentRole-spaceName` función en la lista de funciones de IAM añadidas a tu cuenta.
9. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Para crear y añadir el `CodeCatalyst AWSRoleForCodeCatalystSupport`

1. Antes de empezar en la CodeCatalyst consola, abra el espacio y AWS Management Console, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
3. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de AWS detalles.
4. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM a Amazon CodeCatalyst Space se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Puede que tengas que iniciar sesión para acceder a la página.

5. En los detalles del CodeCatalyst espacio, selecciona Add CodeCatalyst Support role. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza para el rol de desarrollo preliminar. El rol tendrá un nombre `AWSRoleForCodeCatalystSupport` con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Entender la función AWSRoleForCodeCatalystSupport de servicio](#).
6. En la página Añadir función para CodeCatalyst Support, deje seleccionada la opción predeterminada y, a continuación, elija Crear función.

7. En Funciones de IAM disponibles para CodeCatalyst, consulta la `CodeCatalystWorkflowDevelopmentRole-spaceName` función en la lista de funciones de IAM añadidas a tu cuenta.
8. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Tras crear tu ID de AWS constructor, crear tu primer espacio y añadir una cuenta, podrás crear un proyecto. Para obtener más información, consulte [Crear un proyecto en Amazon CodeCatalyst](#). Si es la primera vez que lo usas CodeCatalyst, te sugerimos que comiences con [Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles](#).

Aceptar una invitación y crear tu ID de AWS constructor

Puedes registrarte en Amazon CodeCatalyst como parte de la aceptación de una invitación a un proyecto o un espacio. Como parte de la aceptación de la invitación, se te pedirá que crees un ID de AWS constructor. Utilizarás tu ID de AWS constructor para acceder a los recursos de CodeCatalyst.

Tip

Si necesita ayuda adicional, consulte [¿Problemas al registrarte?](#)

Este es un flujo posible para un usuario que comienza CodeCatalyst con una invitación a un proyecto o un espacio.

Saanvi Sarkar es un desarrollador que ha recibido una invitación para unirse a un CodeCatalyst proyecto como administrador de proyectos. Saanvi acepta la invitación, lo que abre la página de inicio de sesión de CodeCatalyst. Elige registrarse y proporciona una dirección de correo electrónico y una contraseña para crear su AWS ID de constructor. Saanvi podrá usar su ID de AWS constructor para iniciar sesión en CodeCatalyst y en otras aplicaciones. Más adelante, podrá editar su perfil para cambiar su dirección de correo electrónico o contraseña de inicio de sesión. Cuando se le pida que elija un alias, Saanvi especificará el `SaanviSarkar` CodeCatalyst alias que se mostrará CodeCatalyst y que otros miembros del proyecto utilizarán para @mention Saanvi. Una vez que se haya registrado, Saanvi también podrá usar sus credenciales de inicio de sesión para otras aplicaciones que usen credenciales de Builder ID. AWS

Al completar el registro, Saanvi se unirá automáticamente al CodeCatalyst proyecto y al espacio especificados en la invitación. La invitación también proporciona permisos predeterminados para sus

funciones en el proyecto y el espacio. En la configuración del proyecto, el alias de Saanvi aparece en la lista de miembros con su rol de proyecto asignado. Para trabajar con los repositorios de código fuente CodeCatalyst, Saanvi dedica un momento a crear un token de acceso personal (PAT). El PAT se utilizará CodeCatalyst para la autenticación cuando se realicen cambios en la fuente o acciones que requieran un token de autenticación.

Cuando Saanvi trabaje en un proyecto, su alias aparecerá en el registro de actividades laborales del proyecto. Los problemas y comentarios de Saanvi mostrarán su alias, y otros miembros del proyecto podrán escribirla como @mention en sus respuestas. Para @mention, otro miembro del proyecto, Saanvi busca su alias en su perfil. CodeCatalyst

Cuando tiene un momento, Saanvi configura su AWS Builder ID para iniciar sesión CodeCatalyst con la autenticación multifactor (MFA). Con la MFA configurada, Saanvi puede iniciar sesión con una combinación de su CodeCatalyst contraseña y un código de acceso o token de una aplicación de autenticación de terceros aprobada. CodeCatalyst

Aceptar una invitación y crear un Builder ID AWS

Cuando te inviten a un proyecto o espacio en Amazon CodeCatalyst, recibirás un correo electrónico de notify@codecatalyst.aws pidiéndote que aceptes la invitación. Si ya tienes un ID de AWS constructor y has iniciado sesión en él CodeCatalyst, si seleccionas Aceptar la invitación, se abrirá automáticamente el proyecto o el espacio en una pestaña del navegador. Si no has iniciado sesión en la consola pero tienes un AWS Builder ID, accederás a la página de inicio de sesión. Para obtener más información, consulte [Inicia sesión con tu ID de AWS constructor](#).

Si no tienes un AWS Builder ID, si seleccionas Aceptar la invitación, accederás a la página de inicio de sesión, donde deberás elegir la opción de crear tu AWS Builder ID.

Para aceptar una invitación y crear un ID de AWS constructor

1. En el correo electrónico de invitación, selecciona Aceptar invitación.
2. En la página de inicio de sesión, selecciona ¿No te has registrado? Crea tu ID de AWS constructor.

Tip

Tu ID de AWS constructor es una identidad que creas para iniciar sesión. No es lo mismo que unCuenta de AWS.

3. En la página Cree su ID de AWS constructor, en Dirección de correo electrónico, introduzca la dirección de correo electrónico que desee utilizar como ID de AWS constructor.

En Su nombre, introduzca el nombre y apellidos que desee que aparezcan en las aplicaciones en las que utilice su ID de AWS constructor. Se permiten espacios. Este será el nombre de tu perfil de AWS Builder ID, como Mary Major. Puedes cambiar el nombre más adelante.

Elija Siguiente.

Se enviará un código de verificación al correo electrónico que especificó. Introduce este código en Código de verificación y, a continuación, selecciona Verificar. Si no recibes el código después de 5 minutos y no lo encuentras en las carpetas de correo no deseado o basura, selecciona Reenviar el código.

4. Una vez verificado el código, introduce una contraseña que cumpla con los requisitos de Contraseña y Confirmar contraseña.
5. Selecciona Crear ID de AWS constructor.
6. En la página Crea tu alias, introduce el alias que quieras usar como identificador de usuario único CodeCatalyst. Elija una versión abreviada de su nombre sin espacios, como MaryMajor. Otros CodeCatalyst usuarios la usarán para escribirte como @mention en sus comentarios y solicitudes de cambios. Tu CodeCatalyst perfil contendrá tu nombre completo de tu ID de AWS creador y tu CodeCatalyst alias. No puedes cambiar tu CodeCatalyst alias.

Tu nombre completo y tu alias aparecerán en diferentes áreas de CodeCatalyst. Por ejemplo, el nombre de tu perfil aparece para la actividad que aparece en la lista de actividades, pero los miembros del proyecto usarán tu alias como @mention you.

Selecciona Crear alias. Se te dirigirá al proyecto o espacio al que te invitaron.

Inicia sesión con tu ID de AWS constructor

Sigue estos pasos para iniciar sesión en tu CodeCatalyst perfil de Amazon.

Note

¿Ya ha registrado un dispositivo para la autenticación multifactor (MFA)? Le recomendamos encarecidamente que configure la MFA en Amazon CodeCatalyst para aumentar la

seguridad. Para obtener más información, consulte [¿Cómo registrar un dispositivo para su uso con la autenticación multifactorial.](#)

Para iniciar sesión con tu ID de AWS constructor

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Introduzca su dirección de correo electrónico. Si lo desea, elija Guardar mi dirección de correo electrónico si desea guardar su dirección de correo electrónico para futuros inicios de sesión. Elija Continue (Continuar).
3. Introduce tu contraseña. Elija Sign in (Iniciar sesión). Si no recuerdas tu contraseña, sigue los pasos que se indican [He olvidado mi contraseña.](#)

Dispositivos de confianza

Cuando selecciones la opción Este es un dispositivo de confianza en la página de inicio de sesión, Amazon CodeCatalyst considerará que todos los inicios de sesión futuros desde ese dispositivo están autorizados. Amazon CodeCatalyst ofrecerá la opción de introducir un código MFA mientras utilices ese dispositivo de confianza. Algunas excepciones incluyen iniciar sesión desde un navegador nuevo o cuando a tu dispositivo se le haya emitido una dirección IP desconocida.

Aceptar una invitación por correo electrónico para iniciar sesión con SSO

Los usuarios que se añadan a un espacio que admite la federación de identidades reciben un correo electrónico con el enlace al portal de inicio de sesión y la información de configuración. Sigue estos pasos para aceptar una invitación e iniciar sesión con el inicio de sesión único.

Para iniciar sesión con tu ID de AWS constructor en su lugar, consulta [Inicia sesión con tu ID de AWS constructor.](#)

Para aceptar el inicio de sesión único e iniciar sesión con él

1. Selecciona el botón del correo electrónico que acepta la solicitud. Usa el enlace que aparece en el correo electrónico para ir al portal de inicio de sesión del espacio asociado a tu empresa.
2. En Nombre de usuario y contraseña, introduce tus credenciales.

3. Elija Sign in (Iniciar sesión).

Iniciar sesión con SSO

Sigue estos pasos para usar el SSO para iniciar sesión en Amazon CodeCatalyst.

Si prefieres iniciar sesión con tu ID de AWS constructor, consulta [Inicia sesión con tu ID de AWS constructor](#).

Para iniciar sesión con SSO

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En Elegir una opción de inicio de sesión, selecciona Usar inicio de sesión único (SSO).
3. En el nombre de la aplicación de AWS Identity Center, introduzca el nombre de la aplicación proporcionado por el administrador de la federación de identidades.
4. Seleccione Continuar al centro de identidades de IAM.

Ver todos los espacios y proyectos de un usuario

Puede ver una lista de sus espacios y proyectos en la página de inicio del usuario. La página de inicio del usuario muestra una lista de cada espacio al que pertenece el usuario, la función del usuario en ese espacio, como el de administrador del espacio, y los proyectos de cada espacio al que pertenece el usuario.

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el navegador, introduzca la siguiente dirección: <https://codecatalyst.aws/home>

The screenshot displays the Amazon CodeCatalyst interface. At the top, there is a dark header bar with the text 's spaces (9)' on the left, a 'Manage AWS Builder ID' button with an external link icon in the center, and a 'Create space' button on the right. Below the header is a search bar with the placeholder text 'Filter spaces'. The main content area is divided into three sections, each representing a different space:

- EnchantedForest**: A space administrator. It has a 'Create project' button. Below it, there are two project cards: 'WildWaves' and 'FracturedFairyTales'. Each card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.
- org**: A space administrator. It has a 'Create project' button. Below it, there are four project cards: 'migration', 'test', and '12597'. Each card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.
- AnyCompany**: A space member. It has a 'Create project' button. Below it, there is one project card: 'newproject', which lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.

3. Elija el espacio o el proyecto que desee abrir. Si no ves el espacio o el proyecto que esperabas ver, es posible que tengas que iniciar sesión como otro usuario.

Visualización y administración de CodeCatalyst perfiles

Puedes ver los perfiles de usuario en Amazon CodeCatalyst para obtener información como direcciones de correo electrónico y CodeCatalyst alias. También puedes actualizar tu perfil y tu ID de AWS constructor. Si olvidas tu contraseña, puedes solicitar que se restablezca la contraseña.

Ver tu CodeCatalyst perfil

Al registrarte, proporcionas información que se utilizará como credenciales para iniciar sesión en Amazon CodeCatalyst y que se gestionará en tu perfil. Esto incluye tu nombre, apodo y la dirección de correo electrónico que utilizas para iniciar sesión. CodeCatalyst

Note

El apodo de AWS Builder ID no es tu CodeCatalyst alias. Seleccionaste tu CodeCatalyst alias al registrarte.

Para ver tu perfil CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Mi configuración. Se abre la página CodeCatalyst Mis ajustes.
3. Para actualizar tu dirección de correo electrónico o contraseña de AWS Builder ID, o para configurar la MFA, selecciona Administrar AWS Builder ID. Se abre la página AWS Builder ID.

Ver el CodeCatalyst perfil de otro usuario

Para ver el CodeCatalyst perfil de otro usuario

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de navegación lateral, selecciona Configuración del proyecto. Selecciona la pestaña Miembros. Consulta la lista de miembros de tu CodeCatalyst proyecto.
3. Elige el nombre del miembro que quieres buscar o @mention. La página Mi configuración muestra el alias, la dirección de correo electrónico y el nombre completo del usuario. Usa el CodeCatalyst alias para los miembros del proyecto @mention.

Note

El apodo de AWS Builder ID de un usuario no es su CodeCatalyst alias. Seleccionaron su CodeCatalyst alias al registrarse.

Para ver el perfil de otro usuario en tu proyecto, elige su nombre en la lista.

Actualización de tu perfil

En CodeCatalyst, tu perfil está compuesto por información personal gestionada por AWSBuilder ID y ajustes gestionados en CodeCatalyst.

- AWSBuilder ID administra el nombre completo, la dirección de correo electrónico y la contraseña de su perfil. Introdujiste esta información cuando te registraste. Cuando configura MFA para usar una aplicación de autenticación para iniciar sesión en la aplicación, lo CodeCatalyst lleva a la AWS página Builder ID.
- CodeCatalyst La configuración de su token de acceso personal (PAT), CodeCatalyst las notificaciones y las preferencias de idioma se administran en la página Mi configuración de CodeCatalyst Para obtener más información, consulte [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#).

Note

Puedes actualizar tu nombre completo (nombre para CodeCatalyst mostrar) y tu nombre de pila de AWS Builder ID. Sin embargo, no puedes cambiar tu CodeCatalyst alias.

Actualizar tu ID de AWS constructor o tu dirección de correo electrónico

Para actualizar tu dirección ID de creador de AWS o tu dirección de correo electrónico

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Mi configuración. Se abre la página CodeCatalyst Mis ajustes.
3. En la página de perfil, selecciona Administrar ID de AWS Builder. Se abre la página AWSBuilder ID.
4. En la parte izquierda de la página, selecciona Mis datos.
5. En Información de perfil, selecciona Editar para actualizar tu nombre o apodo. Si no especificó ningún apodo, el campo Apodo refleja el nombre del nombre completo. No es tu CodeCatalyst alias.

Note

Esto actualiza el nombre completo y el nombre del AWS Builder ID. Esto no actualiza tu CodeCatalyst alias.

En Información de contacto, selecciona Editar para actualizar tu dirección de correo electrónico.

Note

Esto actualiza la dirección de correo electrónico que utilizarás para iniciar sesión CodeCatalyst.

Cambiar la CodeCatalyst contraseña

Para cambiar la CodeCatalyst contraseña

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Perfil de usuario. Se abre la página CodeCatalyst Mis ajustes.
3. En la página de perfil, selecciona Administrar ID de AWS Builder. Se abre la página AWS Builder ID.
4. En la parte izquierda de la página, selecciona Seguridad.
5. Selecciona Cambiar contraseña y sigue las instrucciones.

Configuración para usar el AWS CLI con CodeCatalyst

La CodeCatalyst consola de Amazon es donde trabajarás en la mayoría de tus tareas diarias. Sin embargo, es posible que desees instalarlas y AWS CLI configurarlas cuando trabajes con entornos de desarrollo, tokens de acceso personal o registros de eventos CodeCatalyst. Debe instalar AWS CLI y configurar un perfil para poder usarlo con él CodeCatalyst.

Para configurar el AWS CLI formulario CodeCatalyst

1. Instale la versión más reciente de AWS CLI. Si ya tiene AWS CLI instalada una versión del, asegúrese de que sea reciente e incluya los comandos correspondientes CodeCatalyst, y actualícela si es necesario. Para comprobar que tiene instalada una versión que incluye CodeCatalyst comandos, abra una línea de comandos y ejecute el siguiente comando:

```
aws codecatalyst help
```

Si ve una lista de CodeCatalyst comandos, tiene una versión compatible CodeCatalyst. Si no se reconoce el comando, actualice su versión AWS CLI a la versión más reciente. Para obtener más información, consulte [Instalación o actualización de la última versión de AWS CLI en la Guía del AWS Command Line Interface usuario](#).

2. Ejecute el `aws configure` comando para crear un perfil si no tiene uno o si desea usar un perfil específico para él CodeCatalyst. Le recomendamos que cree un perfil con nombre para usarlo específicamente CodeCatalyst, pero también puede usar el perfil predeterminado. Para obtener más información, consulte [Conceptos básicos de configuración](#).
3. Edite el `config` archivo del perfil para añadir una sección a la que conectarse de la CodeCatalyst siguiente manera. El archivo `config` se encuentra en `~/.aws/config` en Linux o macOS, o en `C:\Users\USERNAME\.aws\config` en Windows.

```
[profile codecatalyst]
region = us-west-2
sso_session = codecatalyst

[sso-session codecatalyst]
sso_region = us-east-1
sso_start_url = https://view.awsapps.com/start
sso_registration_scopes = codecatalyst:read_write
```

4. Guarde el archivo.
5. Antes de intentar ejecutar cualquier CodeCatalyst comando, abra una nueva terminal o línea de comandos y ejecute el siguiente comando para solicitar y recuperar las credenciales para ejecutar `aws codecatalyst` comandos. `codecatalyst` Sustitúyalo por el nombre de tu perfil si es necesario.

```
aws sso login --profile codecatalyst
```

Para ver ejemplos de codecatalyst comandos, consulte los temas siguientes:

- [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#)
- [Acceder a los eventos registrados en CodeCatalyst](#)

Tutoriales de introducción

Amazon CodeCatalyst ofrece una serie de plantillas diferentes para ayudarte a empezar con tus proyectos. También puedes optar por empezar con un proyecto vacío y añadirle recursos. Sigue los pasos de estos tutoriales para aprender algunas de las formas en las que puedes trabajar CodeCatalyst.

Si es la primera vez que lo usas CodeCatalyst, te sugerimos que comiences con [Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles](#).

Note

Para seguir estos tutoriales, primero debes completar la configuración. Para obtener más información, consulte [Configuración CodeCatalyst](#).

Temas

- [Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles](#)
- [Tutorial: Empezar con un proyecto vacío y añadir recursos manualmente](#)
- [Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo](#)
- [Tutorial: Creación de una aplicación completa con planos de PDK componibles](#)

Para ver tutoriales adicionales que se centran en áreas funcionales específicas de CodeCatalyst, consulte:

- [Cómo empezar a usar las notificaciones de Slack](#)
- [Introducción a los repositorios de CodeCatalyst código fuente y al modelo de aplicación de una sola página](#)
- [Introducción a los flujos de trabajo en CodeCatalyst](#)
- [Cómo empezar con los planos personalizados](#)
- [Empieza con la guía para desarrolladores de Amazon CodeCatalyst Action](#)

Para ver tutoriales detallados, consulta:

- [Tutorial: Carga artefactos a Amazon S3](#)

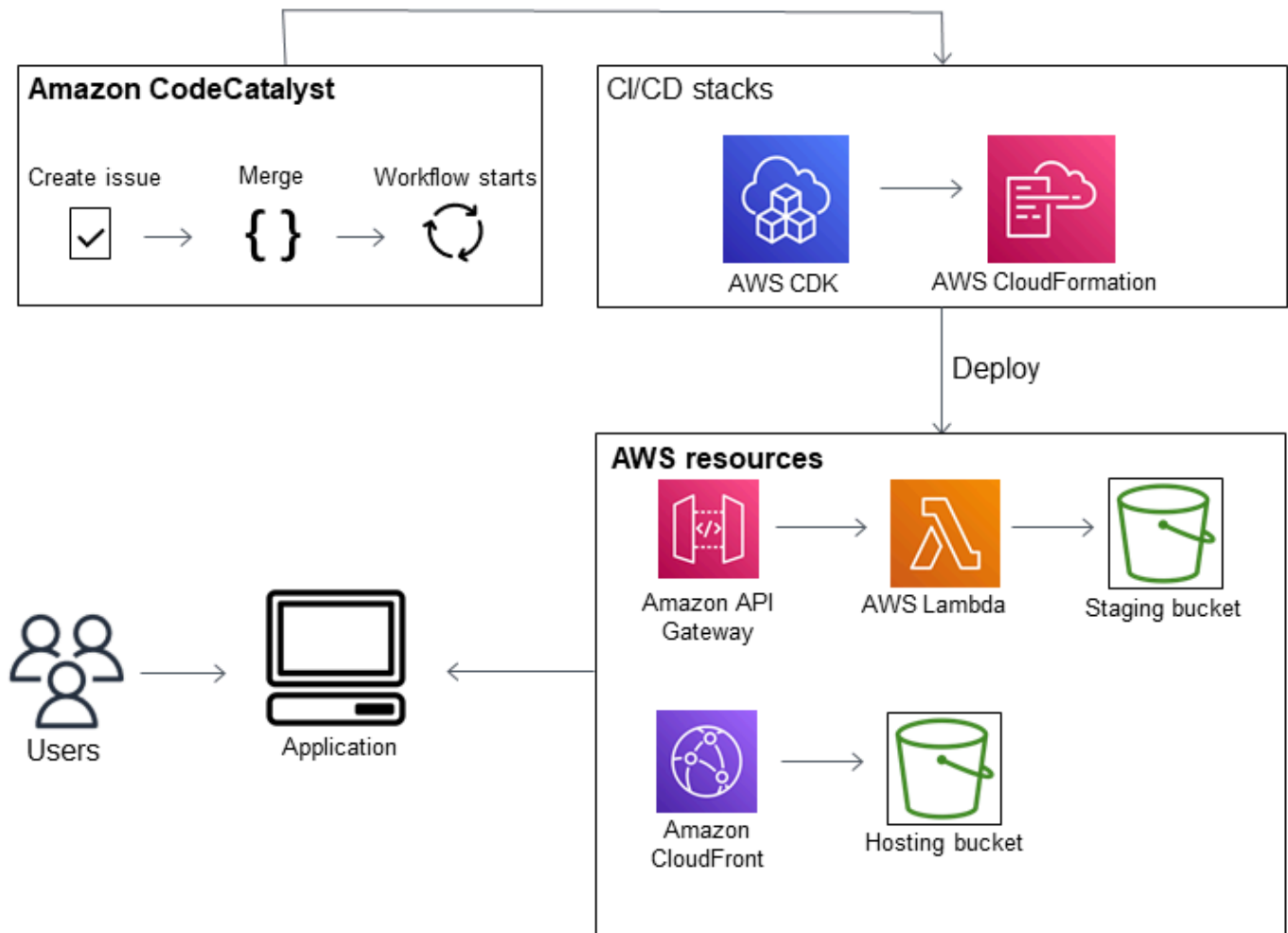
- [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#)
- [Tutorial: Implementación de una aplicación en Amazon ECS](#)
- [Tutorial: Implementación de una aplicación en Amazon EKS](#)
- [Tutorial: código de Lint mediante una acción GitHub](#)
- [Tutorial: Creación y actualización de una aplicación React](#)

Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles

Puedes empezar a desarrollar software de forma más rápida si creas un proyecto con un plano. Un proyecto creado con un plano incluye los recursos que necesita, incluido un repositorio de código fuente para administrar el código y un flujo de trabajo para crear e implementar la aplicación. En este tutorial, te mostraremos cómo usar el modelo de aplicación web moderna de tres niveles para crear un proyecto en Amazon CodeCatalyst. El tutorial también incluye ver el ejemplo implementado, invitar a otros usuarios a trabajar en él y realizar cambios en el código mediante solicitudes de extracción que se crean e implementan automáticamente en los recursos de la red Cuenta de AWS cuando se fusiona la solicitud de extracción. Mientras CodeCatalyst crea su proyecto con informes, fuentes de actividades y otras herramientas, su plan crea AWS recursos en los recursos Cuenta de AWS asociados a su proyecto. Sus archivos de planos le permiten crear y probar una aplicación moderna de muestra e implementarla en la infraestructura de la Nube de AWS.

La siguiente ilustración muestra cómo se utilizan las herramientas de CodeCatalyst para crear un problema con el fin de realizar un seguimiento del cambio, combinarlo y generarlo automáticamente y, a continuación, iniciar un flujo de trabajo en el proyecto de CodeCatalyst que ejecute acciones para permitir AWS CDK y AWS CloudFormation aprovisionar la infraestructura.

Las acciones generan recursos en la función asociada Cuenta de AWS e implementan la aplicación en una AWS Lambda función sin servidor con un punto final de API Gateway. La AWS Cloud Development Kit (AWS CDK) acción convierte una o más AWS CDK pilas en AWS CloudFormation plantillas y despliega las pilas en la Cuenta de AWS. Los recursos de sus paquetes incluyen CloudFront los recursos de Amazon para distribuir contenido web dinámico, una instancia de Amazon DynamoDB para los datos de su aplicación y las funciones y políticas que respaldan la aplicación implementada.



Al crear un proyecto con el esquema de aplicación web moderna de tres niveles, el proyecto se crea con los siguientes recursos:

En el proyecto: CodeCatalyst

- Un [repositorio fuente](#) con código de muestra y flujo de trabajo (YAML)
- Un [flujo de trabajo](#) que compila e implementa el código de muestra cada vez que se realiza un cambio en la rama predeterminada
- Un panel de problemas y un registro de tareas pendientes que puedes usar para planificar y realizar un seguimiento del trabajo
- Un conjunto de informes de pruebas con informes automatizados incluidos en el código de muestra

En el asociado Cuenta de AWS:

- Tres AWS CloudFormation pilas que crean los recursos necesarios para la aplicación.

Para obtener más información sobre los recursos que se crearán en este tutorial AWS y CodeCatalyst como parte de él, consulte [Referencia](#).

Note

Los recursos y los ejemplos incluidos en un proyecto dependen del plan que seleccione. Amazon CodeCatalyst ofrece varios modelos de proyectos que definen los recursos relacionados con su lenguaje o marco definido. Para obtener más información sobre los planos, consulte. [Referencia del plano del proyecto](#)

Temas

- [Requisitos previos](#)
- [Paso 1: Cree un proyecto de aplicación web moderno de tres niveles](#)
- [Paso 2: Invita a alguien a tu proyecto](#)
- [Paso 3: Crea temas para colaborar y realizar un seguimiento del trabajo](#)
- [Paso 4: Visualiza tu repositorio de código fuente](#)
- [Paso 5: Crea un entorno de desarrollo con una rama de prueba y realiza un cambio de código rápido](#)
- [Paso 6: Visualiza el flujo de trabajo que crea la aplicación moderna](#)
- [Paso 7: Pide a otras personas que revisen tus cambios](#)
- [Paso 8: cerrar el problema](#)
- [Eliminar recursos](#)
- [Referencia](#)

Requisitos previos

Para crear un proyecto de aplicación moderno en este tutorial, debe haber completado las tareas de la [Configuración CodeCatalyst](#) siguiente manera:

- Tenga un ID de AWS constructor para iniciar sesión CodeCatalyst.

- Pertenece a un espacio y tienes asignado el rol de administrador del espacio o usuario avanzado en ese espacio. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#), [Gestión de los usuarios del espacio](#) y [Función de administrador de espacios](#).
- Tenga un rol de IAM Cuenta de AWS asociado a su espacio y que tenga el rol de IAM que creó al registrarse. Por ejemplo, durante el registro, tiene la opción de elegir crear un rol de servicio con una política de roles denominada política de roles. CodeCatalystWorkflowDevelopmentRole-*spaceName* El rol tendrá un nombre CodeCatalystWorkflowDevelopmentRole-*spaceName* con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#). Para conocer los pasos necesarios para crear el rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#).

Paso 1: Cree un proyecto de aplicación web moderno de tres niveles

Una vez que lo haya creado, su proyecto es el lugar donde desarrollará y probará el código, coordinará las tareas de desarrollo y verá las métricas del proyecto. El proyecto también contiene las herramientas y los recursos de desarrollo.


En este tutorial, utilizará el modelo de aplicación web moderna de tres niveles para crear una aplicación interactiva. El flujo de trabajo que se crea y ejecuta automáticamente como parte del proyecto creará e implementará la aplicación. El flujo de trabajo solo se ejecuta correctamente una vez que se hayan configurado todos los roles y la información de la cuenta para su espacio. Una vez que el flujo de trabajo se haya ejecutado correctamente, puede visitar la URL del punto final para ver la aplicación.

Para crear un proyecto con un plano

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un proyecto.
3. Elija Crear proyecto.
4. Seleccione Empezar con un plano.
5. En la barra de búsqueda, ingrese **modern**.
6. Seleccione el esquema de aplicación web moderna de tres niveles y, a continuación, elija Siguiente.

7. En **Asigne un nombre a su proyecto**, introduzca un nombre de proyecto. Por ejemplo:

MyExampleProject.

 **Note**

El nombre debe ser único en su espacio.

8. En **Cuenta**, selecciona el Cuenta de AWS que añadiste al registrarte. El plano instalará los recursos en esta cuenta.
9. En **Función de implementación**, elija la función que agregó durante el registro. En este ejemplo, elija `CodeCatalystWorkflowDevelopmentRole-spaceName`.

Si no hay ningún rol en la lista, añada uno. Para añadir un rol, elija **Añadir rol de IAM** y añada el rol al suyo Cuenta de AWS. Para obtener más información, consulte [Cuentas de AWS Administrar un espacio](#).

10. En la plataforma de cómputo, elija **Lambda**.
11. En la opción de alojamiento frontend, elija **Amplify Hosting**. Para obtener más información AWS Amplify, consulte [¿Qué es AWS Amplify el alojamiento?](#) en la Guía AWS Amplify del usuario.
12. En la región de despliegue, introduzca el código de la región en la **Región de AWS** que desea que el plano despliegue la aplicación Mysfits y los recursos de apoyo. Para obtener una lista de los códigos de región, consulte los puntos [finales regionales](#) en. Referencia general de AWS
13. En **Nombre de la aplicación**, deje el valor predeterminado `demysfitsstring`.
14. (Opcional) En **Generar vista previa del proyecto**, seleccione **Ver código** para obtener una vista previa de los archivos fuente que se instalarán en el blueprint. Seleccione **Ver flujo de trabajo** para obtener una vista previa de los archivos de definición del flujo de trabajo de CI/CD que se instalarán en el blueprint. La vista previa se actualiza de forma dinámica en función de sus selecciones.
15. Elija **Crear proyecto**.

El flujo de trabajo del proyecto comienza en cuanto se crea el proyecto. Tardará un poco en terminar de crear e implementar el código. Mientras tanto, continúa e invita a otra persona a tu proyecto.

Paso 2: Invita a alguien a tu proyecto

Ahora que has configurado tu proyecto, invita a otras personas a trabajar contigo.

Para invitar a alguien a tu proyecto

1. Navega hasta el proyecto al que quieres invitar a los usuarios.
2. En el panel de navegación, selecciona Configuración del proyecto.
3. En la pestaña Miembros, elija Invitar.
4. Escribe las direcciones de correo electrónico de las personas a las que quieres invitar como usuarios de tu proyecto. Puedes escribir varias direcciones de correo electrónico separadas por un espacio o una coma. También puedes elegir entre los miembros de tu espacio que no sean miembros del proyecto.
5. Elige el rol del usuario.

Cuando haya terminado de añadir usuarios, elija Invitar.

Paso 3: Crea temas para colaborar y realizar un seguimiento del trabajo

CodeCatalyst te ayuda a realizar un seguimiento de las funciones, las tareas, los errores y cualquier otro trabajo relacionado con tu proyecto que presente problemas. Puedes crear problemas para hacer un seguimiento del trabajo y las ideas necesarias. De forma predeterminada, cuando creas una incidencia, se añade a tu cartera de trabajo pendiente. Puedes mover las incidencias a un foro en el que puedas hacer un seguimiento del trabajo en curso. También puedes asignar una incidencia a un miembro específico del proyecto.

Para crear una incidencia para un proyecto

1. En el panel de navegación, elija Problemas.
2. Selecciona Crear problema.
3. En Título del problema, proporciona un nombre para el problema. Si lo desea, proporcione una descripción del problema. En este ejemplo, utilice **make a change in the src/mysfit_data.json file**.
4. Elija la prioridad, la estimación, el estado y las etiquetas. En Asignatario, selecciona +Agregar para asignarte el problema.
5. Selecciona Crear incidencia. El problema ahora está visible en la pizarra. Elige la carta para mover la emisión a la columna En curso.

Para obtener más información, consulte [Problemas en CodeCatalyst](#).

Paso 4: Visualiza tu repositorio de código fuente

El blueprint instala un repositorio de código fuente que contiene archivos para definir y respaldar la aplicación o el servicio. Algunos directorios y archivos dignos de mención del repositorio de código fuente son:

- `Directorio.cloud9`: contiene archivos auxiliares para el entorno de AWS Cloud9 desarrollo.
- `Directorio.codecatalyst`: contiene el archivo de definición del YAML flujo de trabajo para cada flujo de trabajo incluido en el blueprint.
- `Directorio.idea`: contiene archivos auxiliares para los entornos de desarrollo. JetBrains
- `Directorio.vscode`: contiene archivos auxiliares para el entorno de desarrollo de código de Visual Studio.
- `Directorio.cdkStacks`: contiene los archivos de AWS CDK pila que definen la infraestructura del. Nube de AWS
- `directorio src`: contiene el código fuente de la aplicación.
- `directorio de pruebas`: contiene archivos para las pruebas integrales y unitarias que se ejecutan como parte del flujo de trabajo automatizado de CI/CD que se ejecuta al crear y probar la aplicación.
- `directorio web`: contiene el código fuente de la interfaz. Otros archivos incluyen archivos de proyecto, como el `package.json` archivo que contiene metadatos importantes sobre el proyecto, la `index.html` página del sitio web, el `.eslintrc.cjs` archivo para linting y el `tsconfig.json` archivo para especificar los archivos raíz y las opciones del compilador.
- `Dockerfile` archivo: describe el contenedor de la aplicación.
- `README.md` archivo: contiene información de configuración del proyecto.

Para navegar a los repositorios de origen de un proyecto

1. Navegue hasta su proyecto y realice una de las siguientes acciones:
 - En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio.
 - En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen. En Repositorios de origen, elige el nombre del repositorio de la lista. Puede filtrar la lista de repositorios escribiendo parte del nombre del repositorio en la barra de filtros.

2. En la página de inicio del repositorio, consulta el contenido del repositorio y la información sobre los recursos asociados, como el número de solicitudes de incorporación de datos y los flujos de trabajo. De forma predeterminada, se muestra el contenido de la rama predeterminada. Puede cambiar la vista seleccionando una rama diferente de la lista desplegable.

Paso 5: Crea un entorno de desarrollo con una rama de prueba y realiza un cambio de código rápido

Puedes trabajar rápidamente en el código de tu repositorio de código fuente creando un entorno de desarrollo. Para este tutorial, asumimos que harás lo siguiente:

- Cree un entorno AWS Cloud9 de desarrollo.
- Elija la opción de trabajar en una rama nueva fuera de la rama principal al crear el entorno de desarrollo.
- Usa el nombre `test` de esta nueva rama.

En un paso posterior, utilizarás el entorno de desarrollo para realizar un cambio de código y crear una solicitud de incorporación de cambios.

Para crear un entorno de desarrollo con una nueva rama

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya al proyecto donde desea crear un entorno de desarrollo.
3. Elija el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código, elige Repositorios de código fuente y elige el repositorio para el que quieres crear un entorno de desarrollo.
4. En la página de inicio del repositorio, selecciona Crear entorno de desarrollo.
5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Elija el repositorio que desea clonar, elija Trabajar en una nueva ramificación, introduzca el nombre de la ramificación en el campo Nombre de ramificación, y elija una ramificación a partir de la cual crear la ramificación nueva en el menú desplegable Crear ramificación desde.
7. Si lo desea, añada un alias para el entorno de desarrollo.
8. Si lo desea, elija el botón de editar Configuración del entorno de desarrollo para editar la configuración informática, de almacenamiento o el tiempo de espera del entorno de desarrollo.

9. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo. Se abrirá una nueva pestaña con su entorno de desarrollo en el IDE que elija. Puedes editar el código y confirmar y enviar los cambios.

En esta sección, trabajarás con la aplicación de muestra que has generado realizando cambios en el código CodeCatalyst mediante solicitudes de extracción que se crean e implementan automáticamente en los recursos de la red Cuenta de AWS cuando se fusiona la solicitud de cambios.

Para realizar un cambio en el `src/mysfit_data.json` archivo

1. Navegue hasta el entorno de desarrollo de su proyecto. En AWS Cloud9, expande el menú de navegación lateral para buscar los archivos. `mysfitssrc` Expande y abre `src/mysfit_data.json`.
2. En el archivo, cambie el valor del "Age": campo de 6 a 12. La línea debería tener el siguiente aspecto:

```
{
  "Age": 12,
  "Description": "Twilight's personality sparkles like the night sky and is looking for a forever home with a Greek hero or God. While on the smaller side at 14 hands, he is quite adept at accepting riders and can fly to 15,000 feet. Twilight needs a large area to run around in and will need to be registered with the FAA if you plan to fly him above 500 feet. His favorite activities include playing with chimeras, going on epic adventures into battle, and playing with a large inflatable ball around the paddock. If you bring him home, he'll quickly become your favorite little Pegasus.",
  "GoodEvil": "Good",
  "LawChaos": "Lawful",
  "Name": "Twilight Glitter",
  "ProfileImageUri": "https://www.mythicalmysfits.com/images/pegasus_hover.png",
  "Species": "Pegasus",
  "ThumbImageUri": "https://www.mythicalmysfits.com/images/pegasus_thumb.png"
},
```

3. Guarde el archivo.
4. Cambie al repositorio `mysfits` con el `cd /projects/mysfits` comando.

5. Agrega, confirma y envía tus cambios con los comandos `git add`, `git commit` y `git push`.

```
git add .
git commit -m "make an example change"
git push
```

Paso 6: Visualiza el flujo de trabajo que crea la aplicación moderna

Tras crear el proyecto de aplicación moderna, CodeCatalyst genera varios recursos en su nombre, incluido un flujo de trabajo. Un flujo de trabajo es un procedimiento automatizado definido en un archivo `.yaml` que describe cómo compilar, probar e implementar el código.

En este tutorial, CodeCatalyst creaste un flujo de trabajo y lo iniciaste automáticamente al crear tu proyecto. (Es posible que el flujo de trabajo siga ejecutándose en función de cuánto tiempo haya creado el proyecto). Utilice los siguientes procedimientos para comprobar el progreso del flujo de trabajo, revisar los registros generados y los informes de prueba y, por último, navegar hasta la URL de la aplicación implementada.

Para comprobar el progreso del flujo de trabajo

1. En la CodeCatalyst consola, en el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

Aparece una lista de flujos de trabajo. Estos son los flujos de trabajo que el CodeCatalyst planeó generar e inició al crear el proyecto.

2. Observe la lista de flujos de trabajo. Deberías ver cuatro:
 - Los dos flujos de trabajo de la parte superior corresponden a la `test` rama que creaste anteriormente [Paso 5: Crea un entorno de desarrollo con una rama de prueba y realiza un cambio de código rápido](#). Estos flujos de trabajo son clones de los flujos de trabajo de la `main` sucursal. No `ApplicationDeploymentPipeline` está activo porque está configurado para usarse con la `main` sucursal. El `OnPullRequest` flujo de trabajo no se ejecutó porque no se realizó ninguna solicitud de extracción.
 - Los dos flujos de trabajo de la parte inferior corresponden a la `main` rama que se creó al ejecutar el blueprint anteriormente. El `ApplicationDeploymentPipeline` flujo de trabajo está activo y tiene una ejecución en curso (o finalizada).

Note

Si se produce un `DeployBackenderror` o se produce un error en la `ApplicationDeploymentPipeline` ejecución, es posible que se deba a que anteriormente ejecutó la aplicación web moderna de tres niveles y dejó recursos antiguos que entraban en conflicto con el esquema actual. @cdk_bootstrap Deberá eliminar estos recursos antiguos y, a continuación, volver a ejecutar el flujo de trabajo. Para obtener más información, consulte [Eliminar recursos](#).

3. Selecciona el `ApplicationDeploymentPipeline` flujo de trabajo asociado a la `main` sucursal, en la parte inferior. Este flujo de trabajo se ejecutó utilizando el código fuente de la `main` sucursal.



Aparece un diagrama de flujo de trabajo. El diagrama muestra varios bloques, cada uno de los cuales representa una tarea o acción. La mayoría de las acciones se organizan verticalmente, con las acciones de la parte superior ejecutándose antes que las de abajo. Las acciones que se organizan una al lado de la otra se ejecutan en paralelo. Todas las acciones agrupadas deben ejecutarse correctamente antes de que pueda iniciarse la acción que se encuentra debajo de ellas.

Los bloques principales son:

- `WorkflowSource`— Este bloque representa tu repositorio fuente. Muestra, entre otra información, el nombre del repositorio de origen (`mysfits`) y la confirmación que inició automáticamente la ejecución del flujo de trabajo. CodeCatalyst generaste esta confirmación cuando creaste tu proyecto.
 - `Construir`: este bloque representa un grupo de dos acciones que deben completarse correctamente para que comience la siguiente.
 - `DeployBackend`— Este bloque representa una acción que despliega los componentes de backend de la aplicación en la nube. AWS
 - `Pruebas`: este bloque representa un grupo de dos acciones de prueba que deben completarse correctamente para que se inicie la siguiente acción.
 - `DeployFrontend`— Este bloque representa una acción que despliega los componentes de la interfaz de usuario de la aplicación en la nube. AWS
4. Seleccione la pestaña `Definición` (cerca de la parte superior). El [archivo de definición del flujo](#) de trabajo aparece a la derecha. El archivo tiene las siguientes secciones destacables:

- Una `Triggers` sección, en la parte superior. En esta sección se indica que el flujo de trabajo debe iniciarse siempre que el código se envíe a la `main` rama del repositorio de origen. Si se envía a otras ramas (por ejemplo `test`), no se iniciará este flujo de trabajo. El flujo de trabajo se ejecuta con los archivos de la `main` rama.
 - Una `Actions` sección, debajo `Triggers`. En esta sección se definen las acciones que se muestran en el diagrama de flujo de trabajo.
5. Seleccione la pestaña Estado más reciente (situada cerca de la parte superior) y elija cualquier acción del diagrama de flujo de trabajo.
 6. A la derecha, seleccione la pestaña Configuración para ver los ajustes de configuración utilizados por la acción durante la última ejecución. Cada parámetro de configuración tiene una propiedad coincidente en el archivo de definición del flujo de trabajo.
 7. Deje la consola abierta y continúe con el siguiente procedimiento.

Para revisar los registros de compilación y los informes de prueba

1. Seleccione la pestaña Estado más reciente.
2. En el diagrama de flujo de trabajo, elija la `DeployFrontend` acción.
3. Espere a que finalice la acción. Esté atento a que el icono «en curso» () cambie a un icono de «éxito» (.
4. Elige la acción `build_backend`.
5. Seleccione la pestaña Registros y expande un par de secciones para ver los mensajes de registro de estos pasos. Puedes ver los mensajes relacionados con la configuración del backend.
6. Seleccione la pestaña Informes y, a continuación, seleccione el `backend-coverage.xml` informe. CodeCatalyst muestra el informe asociado. El informe muestra las pruebas de cobertura de código que se realizaron e indica la proporción de líneas de código que se validaron correctamente mediante las pruebas, como el 80%.

Para obtener más información sobre los informes de las pruebas, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#).

i Tip

También puede ver los informes de las pruebas seleccionando Informes en el panel de navegación.

7. Deje la CodeCatalyst consola abierta y continúe con el siguiente procedimiento.

Para confirmar que la aplicación moderna se implementó correctamente

1. Vuelva al ApplicationDeploymentPipelineflujo de trabajo y elija el enlace Ejecutar **cadena** de la última ejecución.
2. En el diagrama de flujo de trabajo, busca la DeployFrontendacción y selecciona el enlace Ver aplicación. Aparece el sitio web de Mysfit.

i Note

Si no ves el enlace Ver la aplicación dentro de la DeployFrontendacción, asegúrate de elegir el enlace del ID de ejecución.

3. Busca el pegaso Mysfit llamado Twilight Glitter. Anote el valor de la edad. Lo es6. Harás un cambio de código para actualizar la edad.

Paso 7: Pide a otras personas que revisen tus cambios

Ahora que tienes los cambios en una sucursal llamada `test`, puedes pedir a otras personas que los revisen creando una solicitud de cambios. Realiza los siguientes pasos para crear una solicitud de extracción que combine los cambios de la `test` sucursal en la `main` sucursal.

Para crear una solicitud de extracción

1. Vaya a su proyecto.
2. Realice una de las acciones siguientes:
 - En el panel de navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, selecciona Crear solicitud de extracción.
 - En la página de inicio del repositorio, selecciona Más y, a continuación, selecciona Crear solicitud de extracción.


- En la página del proyecto, selecciona Crear solicitud de extracción.
3. En el repositorio de fuentes, asegúrate de que el repositorio de fuentes especificado es el que contiene el código confirmado. Esta opción solo aparece si no creaste la solicitud de extracción desde la página principal del repositorio.
 4. En la rama de destino, elige la rama en la que deseas combinar el código después de revisarlo.
 5. En la rama Fuente, elige la rama que contiene el código confirmado.
 6. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios a entender qué es lo que hay que revisar y por qué.
 7. (Opcional) En la descripción de la solicitud de retirada, proporciona información como un enlace a los problemas o una descripción de tus cambios.

 Tip

Puedes elegir Escribir una descripción para que yo genere CodeCatalyst automáticamente una descripción de los cambios incluidos en la solicitud de cambios. Puedes realizar cambios en la descripción generada automáticamente después de añadirla a la solicitud de extracción.

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

8. (Opcional) En Problemas, selecciona Vincular problemas y, a continuación, elige un problema de la lista o introduce su ID. Para desvincular una incidencia, selecciona el icono de desvinculación.
9. (Opcional) En Revisores obligatorios, selecciona Agregar revisores obligatorios. Elija de la lista de miembros del proyecto para añadirlos. Los revisores necesarios deben aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.

 Note

No puedes añadir un revisor como revisor obligatorio y como revisor opcional. No puedes añadirte a ti mismo como revisor.

10. (Opcional) En Revisores opcionales, selecciona Añadir revisores opcionales. Elija de la lista de miembros del proyecto para añadirlos. Los revisores opcionales no tienen que aprobar los cambios como requisito para poder fusionar la solicitud de extracción en la sucursal de destino.

11. Revisa las diferencias entre las sucursales. La diferencia que se muestra en una solicitud de extracción son los cambios entre la revisión en la rama de origen y la base de fusión, que es la confirmación principal de la rama de destino en el momento en que se creó la solicitud de extracción. Si no se muestra ningún cambio, es posible que las ramas sean idénticas o que hayas elegido la misma rama tanto para el origen como para el destino.
12. Cuando estés seguro de que la solicitud de extracción contiene el código y los cambios que deseas revisar, selecciona Crear.

Note

Después de crear la solicitud de extracción, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a recursos, como archivos, utilizando el signo @ seguido del nombre del archivo.

Al crear la solicitud de extracción, el OnPullRequestflujo de trabajo comienza a utilizar los archivos fuente de la test sucursal. Mientras tus revisores aprueban el cambio de código, puedes observar los resultados seleccionando el flujo de trabajo y viendo el resultado de la prueba.

Una vez que hayas revisado el cambio, puedes combinar el código. Al combinar el código con la rama predeterminada, se iniciará automáticamente el flujo de trabajo que generará e implementará los cambios.

Para fusionar una solicitud de extracción desde la consola CodeCatalyst

1. Navega hasta tu proyecto de aplicación moderno.
2. En la página del proyecto, en Abrir solicitudes de extracción, elige la solicitud de extracción que deseas fusionar. Si no ves la solicitud de extracción, selecciona Ver todas y, a continuación, selecciónala de la lista. Elija Merge (Combinar).
3. Elige entre las estrategias de fusión disponibles para la solicitud de extracción. Si lo deseas, selecciona o deselecciona la opción para eliminar la rama de origen después de fusionar la solicitud de extracción y, a continuación, selecciona Fusionar.

Note

Si el botón Fusionar no está activo o si aparece la etiqueta No se puede combinar, significa que uno o varios revisores necesarios aún no han aprobado la solicitud de

extracción o la solicitud de extracción no se puede fusionar en la consola. CodeCatalyst Un revisor que no ha aprobado una solicitud de extracción se indica con un icono de reloj en la sección Información general, en el área de detalles de la solicitud de extracción. Si todos los revisores necesarios han aprobado la solicitud de extracción, pero el botón Combinar sigue sin estar activo, es posible que se esté produciendo un conflicto de fusión. Puedes resolver los conflictos de fusión para la rama de destino en la CodeCatalyst consola y, a continuación, fusionar la solicitud de extracción, o puedes resolver los conflictos y fusionar localmente y, a CodeCatalyst continuación, enviar la confirmación que contiene la fusión. Para obtener más información, consulta [Fusionar una solicitud de cambios \(Git\)](#) la documentación de Git.

Una vez que hayas fusionado los cambios de una test rama a otra **main**, el cambio inicia automáticamente el ApplicationDeploymentPipeline flujo de trabajo que crea y despliega el cambio.

Para ver cómo se ejecuta la confirmación combinada en el flujo de trabajo ApplicationDeploymentPipeline

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. En Flujos de trabajo, en ApplicationDeploymentPipeline, expanda Ejecuciones recientes. Puede ver la ejecución del flujo de trabajo iniciada por la confirmación de fusión. Si lo desea, selecciónelo para ver el progreso de la ejecución.
3. Cuando se complete la ejecución, vuelve a cargar la URL que visitaste anteriormente. Observa el pegaso para comprobar que la edad ha cambiado.



Paso 8: cerrar el problema

Cuando se resuelve un problema, se puede cerrar en la CodeCatalyst consola.

Para cerrar un problema de un proyecto

1. Vaya a su proyecto.
2. En el panel de navegación, elija Problemas.
3. rag-and-drop Dirija el problema a la columna Listo.

Para obtener más información, consulte [Problemas en CodeCatalyst](#).

Eliminar recursos

Limpie CodeCatalyst y elimine AWS los rastros de este tutorial de su entorno.

Puede optar por seguir utilizando el proyecto que utilizó para este tutorial o puede eliminar el proyecto y sus recursos asociados.

Note

Al eliminar este proyecto, se eliminarán todos los repositorios, números y artefactos del proyecto para todos los miembros.

Para eliminar un proyecto

1. Navega hasta tu proyecto y, a continuación, selecciona la configuración del proyecto.
2. Elija la pestaña General.
3. En el nombre del proyecto, selecciona Eliminar proyecto.

Para eliminar recursos en Amazon S3 AWS CloudFormation y en Amazon S3

1. Inicia sesión en el AWS Management Console con la misma cuenta que añadiste a tu CodeCatalyst espacio.
2. Ve al AWS CloudFormation servicio.
3. Elimine la pila de **cadena**s mysfits.
4. **Elimine la pila de cadenas de development-mysfits.**
5. Elija (pero no elimine) la pila de CDKToolkit. Elija la pestaña Recursos. Elija el StagingBucket enlace y elimine el depósito y su contenido en Amazon S3.

Note

Si no elimina este depósito manualmente, es posible que aparezca un error al volver a ejecutar el blueprint de la aplicación web moderna de tres niveles.


6. (Opcional) Elimine la pila de CDKToolkit.

Referencia

El moderno plan de aplicación web de tres niveles despliega los recursos en su CodeCatalyst espacio y en su AWS cuenta en la nube. AWS Estos recursos son:

- En tu CodeCatalyst espacio:
 - Un CodeCatalyst proyecto que incluye los siguientes recursos:

- Un [repositorio de código fuente](#): este repositorio contiene código de muestra para una aplicación web «Mysfits».
 - Un [flujo de trabajo](#): este flujo de trabajo crea e implementa el código de la aplicación Mysfits cada vez que se realiza un cambio en la rama predeterminada
 - Un [panel de problemas](#) y un registro de trabajo pendiente: este panel y el trabajo pendiente se pueden utilizar para planificar y realizar un seguimiento del trabajo.
 - Un [conjunto de informes de pruebas](#): este conjunto incluye informes automatizados incluidos en el código de muestra.
- En el asociado Cuenta de AWS:
 - Una pila de CDKToolkit: esta pila implementa los siguientes recursos:
 - Un depósito provisional de Amazon S3, una política del depósito y la AWS KMS clave utilizada para cifrar el depósito.
 - Una función de implementación de IAM para la acción de implementación.
 - AWS Funciones y políticas de IAM en apoyo de los recursos de la pila.

 Note

El CDKToolkit no se desmonta ni se vuelve a crear para cada implementación. Se trata de una pila que se inicia en cada cuenta para respaldar la. AWS CDK

- Una pila de **cadena**s mysfits de desarrollo: esta BackEnd pila implementa los siguientes recursos de back-end:
 - Un punto final de Amazon API Gateway.
 - AWS Funciones y políticas de IAM que respaldan los recursos de la pila.
 - Una AWS Lambda función y una capa proporcionan la plataforma de cómputo sin servidor para la aplicación moderna.
 - Una política y una función de IAM para el despliegue del bucket y la función Lambda.
- Una pila de **cadena**s mysfits: esta pila implementa la aplicación de interfaz. AWS Amplify

Véase también

Para obtener más información sobre los AWS servicios en los que se crean los recursos como parte de este tutorial, consulte lo siguiente:

- Amazon S3: un servicio para almacenar sus activos de front-end en un servicio de almacenamiento de objetos que ofrece escalabilidad, alta disponibilidad de datos, seguridad y rendimiento líderes del sector. Para obtener más información, consulte la [Guía del usuario de Amazon S3](#).
- Amazon API Gateway: un servicio para crear, publicar, mantener, supervisar y proteger REST, HTTP y WebSocket API a cualquier escala. Para obtener más información, consulte la [Guía para desarrolladores de API Gateway](#).
- Amplify: un servicio para alojar su aplicación frontend. Para obtener más información, consulte la Guía del [usuario AWS Amplify de hosting](#).
- AWS Cloud Development Kit (AWS CDK)— Un marco para definir la infraestructura de nube en el código y aprovisionarla mediante AWS CloudFormation ella. AWS CDK Incluye el AWS CDK kit de herramientas, que es una herramienta de línea de comandos para interactuar con AWS CDK aplicaciones y pilas. Para obtener información, consulte la [Guía para desarrolladores de AWS Cloud Development Kit \(AWS CDK\)](#).
- Amazon DynamoDB: un servicio de base de datos NoSQL totalmente gestionado para almacenar datos. Para obtener más información, consulte la [Guía para desarrolladores de Amazon DynamoDB](#).
- AWS Lambda— Un servicio para invocar el código en una infraestructura informática de alta disponibilidad sin aprovisionar ni administrar servidores. Para obtener información, consulte la [Guía para desarrolladores de AWS Lambda](#).
- AWS IAM: un servicio para controlar de forma segura el acceso a los recursos AWS y sus recursos. Para obtener más información, consulte la [Guía del usuario de IAM](#).

Tutorial: Empezar con un proyecto vacío y añadir recursos manualmente

Para crear un proyecto vacío sin ningún recurso predefinido en su interior, elija el esquema de proyecto vacío al crear el proyecto. Tras crear un proyecto vacío, puede crearlo y añadirle recursos según las necesidades del proyecto. Como los proyectos creados sin un esquema están vacíos al crearlos, esta opción requiere más conocimientos sobre la creación y la configuración de CodeCatalyst recursos para empezar.

Temas

- [Requisitos previos](#)

- [Cree un proyecto vacío](#)
- [Crea un repositorio de origen](#)
- [Crea un flujo de trabajo para crear, probar e implementar un cambio de código](#)
- [Invita a alguien a tu proyecto](#)
- [Crea temas en los que colaborar y hacer un seguimiento del trabajo](#)

Requisitos previos

Para crear un proyecto vacío, debe tener asignado el rol de administrador de Space o usuario avanzado. Si es la primera vez que inicia sesión CodeCatalyst, consulte [Configuración CodeCatalyst](#).

Cree un proyecto vacío

Crear un proyecto es el primer paso para poder trabajar juntos. Si quieres crear tus propios recursos, como repositorios de fuentes y flujos de trabajo, puedes empezar con un proyecto vacío.

Para crear un proyecto vacío

1. Vaya al espacio en el que desea crear un proyecto.
2. En el panel de control del espacio, seleccione Crear proyecto.
3. Elija Empezar desde cero.
4. En Asigne un nombre a su proyecto, introduzca el nombre que desea asignar a su proyecto. El nombre debe ser único dentro de su espacio.
5. Elija Crear proyecto.

Ahora que tienes un proyecto vacío, el siguiente paso es crear un repositorio de origen.

Crea un repositorio de origen

Crea un repositorio de código fuente para almacenar el código de tu proyecto y colaborar en él. Los miembros del proyecto pueden clonar este repositorio en sus ordenadores locales para trabajar con el código. Como alternativa, puede optar por vincular un repositorio alojado en un servicio compatible, pero esto no se trata en este tutorial. Para obtener más información, consulte [Vincular un repositorio de fuentes](#).

Para crear un repositorio de origen

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su proyecto.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
5. En Nombre del repositorio, especifique un nombre para el repositorio. En esta guía, utilizamos *codecatalyst-source-repository*, pero puede elegir un nombre diferente. Los nombres de repositorio deben ser únicos dentro de un proyecto. Para obtener más información sobre los requisitos de los nombres de los repositorios, consulte [Cuotas para los repositorios de fuentes en CodeCatalyst](#).
6. (Opcional) En la descripción, añada una descripción del repositorio que ayude a los demás usuarios del proyecto a entender para qué se utiliza el repositorio.
7. (Opcional) Agrega un `.gitignore` archivo para el tipo de código que planeas insertar.
8. Seleccione Crear.

Note

CodeCatalyst agrega un README .md archivo a tu repositorio al crearlo. CodeCatalyst también crea una confirmación inicial para el repositorio en una rama predeterminada llamada main. Puede editar o eliminar el archivo README.md, pero no puede cambiar ni eliminar la rama predeterminada.

Puedes añadir código rápidamente a tu repositorio creando un entorno de desarrollo. Para este tutorial, le recomendamos que cree un entorno de desarrollo utilizando AWS Cloud9 y elija la opción de crear una rama a partir de la rama principal al crear el entorno de desarrollo. Usamos el nombre **test** para esta rama, pero puedes introducir un nombre de rama diferente si lo prefieres.

Para crear un entorno de desarrollo con una nueva rama

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya al proyecto donde desea crear un entorno de desarrollo.
3. Elija el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código, elige Repositorios de código fuente y elige el repositorio para el que quieres crear un entorno de desarrollo.

4. En la página de inicio del repositorio, selecciona Crear entorno de desarrollo.
5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Elija el repositorio que desea clonar, elija Trabajar en una nueva ramificación, introduzca el nombre de la ramificación en el campo Nombre de ramificación, y elija una ramificación a partir de la cual crear la ramificación nueva en el menú desplegable Crear ramificación desde.
7. Si lo desea, añada un alias para el entorno de desarrollo.
8. Si lo desea, elija el botón de editar Configuración del entorno de desarrollo para editar la configuración informática, de almacenamiento o el tiempo de espera del entorno de desarrollo.
9. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo. Se abrirá una nueva pestaña con su entorno de desarrollo en el IDE que elija. Puedes editar el código y confirmar y enviar los cambios.

Crea un flujo de trabajo para crear, probar e implementar un cambio de código

En CodeCatalyst, organizas la creación, las pruebas y el despliegue de tus aplicaciones o servicios en flujos de trabajo. Los flujos de trabajo se componen de acciones y se pueden configurar para que se ejecuten automáticamente después de que se produzcan eventos específicos en el repositorio de origen, como la inserción de código o la apertura o actualización de una solicitud de extracción de cambios. Para obtener más información acerca de los flujos de trabajo, consulte [Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst](#).

Sigue las instrucciones que se indican [Introducción a los flujos de trabajo en CodeCatalyst](#) para crear tu primer flujo de trabajo.

Invita a alguien a tu proyecto

Ahora que has configurado tu proyecto personalizado, invita a otras personas a trabajar contigo.

Para invitar a alguien a tu proyecto

1. Navega hasta el proyecto al que quieres invitar a los usuarios.
2. En el panel de navegación, selecciona Configuración del proyecto.
3. En la pestaña Miembros, elija Invitar.

4. Escribe las direcciones de correo electrónico de las personas a las que quieres invitar como usuarios de tu proyecto. Puedes escribir varias direcciones de correo electrónico separadas por un espacio o una coma. También puedes elegir entre los miembros de tu espacio que no sean miembros del proyecto.
5. Elige el rol del usuario.

Cuando haya terminado de añadir usuarios, elija Invitar.

Crea temas en los que colaborar y hacer un seguimiento del trabajo

CodeCatalyst te ayuda a realizar un seguimiento de las funciones, las tareas, los errores y cualquier otro trabajo relacionado con tu proyecto que presente problemas. Puedes crear problemas para hacer un seguimiento del trabajo y las ideas necesarias. De forma predeterminada, cuando creas una incidencia, se añade a tu cartera de trabajo pendiente. Puedes mover las incidencias a un foro en el que puedas hacer un seguimiento del trabajo en curso. También puedes asignar una incidencia a un miembro específico del proyecto.

Para crear una incidencia para un proyecto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.

Asegúrese de navegar por el proyecto en el que desea crear las incidencias. Para ver todos los proyectos, en el panel de navegación, selecciona Amazon y CodeCatalyst, si es necesario, selecciona Ver todos los proyectos. Elige el proyecto en el que quieres crear o trabajar con los problemas.

2. En el panel de navegación, selecciona Track y, a continuación, selecciona Backlog.
3. Selecciona Crear incidencia.
4. En Título del problema, proporciona un nombre para el problema. Si lo desea, proporcione una descripción del problema. Elija el estado, la prioridad y la estimación del problema si lo desea. También puede asignar el problema a un miembro del proyecto de la lista de miembros del proyecto.

Tip

Puedes asignar un problema a Amazon Q para que Amazon Q intente resolverlo. Si el intento tiene éxito, se creará una solicitud de cambios y el estado del problema cambiará a En revisión para que puedas revisar y probar el código. Para obtener más información,

consulte [Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo](#).

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

5. Seleccione Guardar.

Una vez que hayas creado los problemas, puedes asignarlos a los miembros del proyecto, estimarlos y hacer un seguimiento de ellos en un tablero Kanban. Para obtener más información, consulte [Problemas en CodeCatalyst](#).

Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo

Las funciones de IA generativa de Amazon CodeCatalyst se encuentran en versión preliminar y están sujetas a cambios. Solo están disponibles en la región EE.UU. Oeste (Oregón). El acceso a las funciones generativas de IA varía según el nivel. Para más información, consulte [Precios de](#) .

Si tienes un proyecto y un repositorio de código fuente en Amazon CodeCatalyst en un espacio en el que están habilitadas las funciones de IA generativa, puedes utilizarlas para acelerar el desarrollo de software. Los desarrolladores suelen tener más tareas que hacer que tiempo para llevarlas a cabo. A menudo, no se toman el tiempo de explicar los cambios de código a sus compañeros de equipo cuando crean solicitudes de cambios para revisarlos, ya que esperan que los demás usuarios consideren que los cambios se explican por sí mismos. Los creadores y revisores de las solicitudes de extracción tampoco tienen tiempo para buscar y leer detenidamente todos los comentarios de una solicitud de extracción, especialmente si la solicitud de extracción tiene varias revisiones. CodeCatalyst incluye funciones de IA generativa que pueden ayudar a los miembros del equipo a realizar sus tareas con mayor rapidez y aumentar el tiempo que tienen para concentrarse en las partes más importantes de su trabajo.

Note

Desarrollado por Amazon Bedrock: AWS implementa la [detección automática de abusos](#). Como las funciones Escribir descripción para mí, Crear resumen de contenido y Asignar

problemas a Amazon Q están integradas en Amazon Bedrock, los usuarios pueden aprovechar al máximo los controles implementados en Amazon Bedrock para garantizar la protección, la seguridad y el uso responsable de la inteligencia artificial (IA).

En este tutorial, aprenderás a usar las funciones de IA generativa CodeCatalyst para ayudarte a resumir los cambios entre las ramas al crear solicitudes de extracción y a resumir los comentarios dejados en una solicitud de extracción. También aprenderás a crear problemas con tus ideas para realizar cambios o mejoras simples en el código y a asignarlos a Amazon Q.

Requisitos previos

Para trabajar con las CodeCatalyst funciones de este tutorial, primero debe haber completado y tener acceso a los siguientes recursos:

- Tienes un AWS Builder ID o una identidad de inicio de sesión único (SSO) para iniciar sesión. CodeCatalyst
- Tu proyecto se encuentra en un espacio que tiene habilitadas las funciones de IA generativa. Para obtener más información, consulta [Administrar las funciones de IA generativa](#).
- Tienes el rol de colaborador o administrador del proyecto en un proyecto en ese espacio.
- El proyecto tiene al menos un repositorio de origen configurado para él. No se admiten los repositorios enlazados.
- Al asignar problemas a una solución inicial creada por IA generativa, el proyecto no se puede configurar con la extensión de Jira Software. La extensión no es compatible con esta función.

Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#), [Problemas en CodeCatalyst](#), [Extensiones en CodeCatalyst](#) y [Trabajar con roles en Amazon CodeCatalyst](#).

Este tutorial se basa en un proyecto creado utilizando el modelo de aplicación web moderna de tres niveles con Python. Si utilizas un proyecto creado con un esquema diferente, puedes seguir los pasos, pero algunos detalles pueden variar, como el código y el lenguaje de muestra.

Crea un resumen de los cambios de código entre las ramas al crear una solicitud de extracción

Una solicitud de extracción es la principal forma en que tú y otros miembros del proyecto podéis revisar, comentar y combinar los cambios de código de una rama a otra. Puedes usar las solicitudes de extracción para revisar los cambios de código de forma colaborativa para detectar cambios o correcciones menores, incorporaciones de funciones importantes o nuevas versiones del software que hayas publicado. Resumir los cambios en el código y su intención como parte de la descripción de la solicitud de extracción es útil para otras personas que quieran revisar el código y también ayuda a comprender el historial de los cambios introducidos en el código a lo largo del tiempo. Sin embargo, los desarrolladores suelen confiar en su código para explicarse por sí solo o para proporcionar detalles ambiguos en lugar de describir sus cambios con suficiente detalle como para que los revisores entiendan lo que están revisando o cuál era la intención detrás de los cambios en el código.

Puedes usar la función *Escribe una descripción para mí* al crear solicitudes de extracción para que Amazon Q cree una descripción de los cambios contenidos en una solicitud de extracción. Al elegir esta opción, Amazon Q analiza las diferencias entre la rama de origen que contiene los cambios de código y la rama de destino en la que desea combinar estos cambios. A continuación, crea un resumen de cuáles son esos cambios, así como su mejor interpretación de la intención y el efecto de esos cambios.

Puedes probar esta función con cualquier solicitud de cambios que crees, pero en este tutorial la probaremos realizando algunos cambios sencillos en el código contenido en un proyecto creado en un blueprint de una aplicación web moderna de tres niveles basada en Python.

Tip

Si utilizas un proyecto creado con un esquema diferente o con tu propio código, puedes seguir este tutorial, pero los ejemplos de este tutorial no coincidirán con el código de tu proyecto. En lugar del ejemplo que se sugiere a continuación, realiza cambios sencillos en el código de tu proyecto en una rama y, a continuación, crea una solicitud de extracción para probar la función, tal y como se muestra en los pasos siguientes.


En primer lugar, crearás una rama en el repositorio de código fuente. A continuación, realizarás un cambio rápido de código en un archivo de esa rama mediante el editor de texto de la consola. A

continuación, crearás una solicitud de cambios y utilizarás la función *Escribe una descripción para resumir los cambios que has realizado*.

Para crear una sucursal (consola)

1. En la CodeCatalyst consola, navega hasta el proyecto en el que reside tu repositorio de origen.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona *Código* y, a continuación, selecciona *Repositorios de origen*.
3. Elija el repositorio en el que desee crear una rama.
4. En la página de información general del repositorio, selecciona *Más* y, a continuación, selecciona *Crear rama*.
5. Introduzca un nombre para la rama.
6. Elige una rama desde la que crear la rama y, a continuación, selecciona *Crear*.

Una vez que tengas una sucursal, edita un archivo en esa rama con un simple cambio. En este ejemplo, editarás el `test_endpoint.py` archivo para cambiar el número de reintentos de las pruebas de 5 **3** a 5.

 Tip

También puedes optar por crear o usar un entorno de desarrollo para realizar este cambio de código. Para obtener más información, consulte [Creación de un entorno de desarrollo](#).

Para editar el `test_endpoint.py` archivo en la consola

1. En la página de información general del repositorio de **mysfits** origen, seleccione el menú desplegable de ramas y elija la rama que creó en el procedimiento anterior.
2. En *Archivos*, navegue hasta el archivo que desee editar. Por ejemplo, para editar el `test_endpoint.py` archivo, expanda las pruebas, expanda `integ` y, a continuación, elija `test_endpoint.py`.
3. Elija *Editar*.
4. En la línea 7, cambie el número de veces que se van a volver a intentar todas las pruebas:

```
def test_list_all(retry=3):
```

a:

```
def test_list_all(retry=5):
```

5. Selecciona Confirmar y confirma los cambios en tu rama.

Ahora que tienes una sucursal con un cambio, puedes crear una solicitud de cambios.

Crema una solicitud de extracción con un resumen de los cambios

1. En la página de información general del repositorio, selecciona Más y, a continuación, selecciona Crear solicitud de extracción.
2. En la rama de destino, selecciona la rama en la que deseas combinar el código después de revisarlo.

 Tip

Elija la rama desde la que creó la rama en el procedimiento anterior para obtener la demostración más sencilla de esta función. Por ejemplo, si creaste tu sucursal a partir de la rama predeterminada del repositorio, elije esa sucursal como la sucursal de destino para tu solicitud de incorporación de cambios.

3. En la rama Fuente, elige la rama que contiene los cambios que acabas de confirmar en el `test_endpoint.py` archivo.
4. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios a entender qué es lo que hay que revisar y por qué.
5. En la descripción de la solicitud de extracción, selecciona Escribir descripción para CodeCatalyst crear una descripción de los cambios incluidos en la solicitud de extracción.
6. Aparece un resumen de los cambios. Revisa el texto sugerido y, a continuación, selecciona Aceptar y añadir a la descripción.
7. Si lo desea, modifique el resumen para que refleje mejor los cambios que ha realizado en el código. También puedes añadir revisores o vincular los problemas a este pull request. Cuando hayas terminado de realizar los cambios adicionales que desees, selecciona Crear.

Crea un resumen de los comentarios dejados sobre los cambios de código en una solicitud de extracción

Cuando los usuarios revisan una solicitud de extracción, suelen dejar varios comentarios sobre los cambios en esa solicitud de extracción. Si hay muchos comentarios de muchos revisores, puede resultar difícil elegir temas comunes en los comentarios o incluso asegurarte de que has revisado todos los comentarios de todas las revisiones. Puedes usar la función Crear resumen de comentarios para que Amazon Q analice todos los comentarios dejados sobre los cambios de código en una solicitud de cambios de código y cree un resumen de esos comentarios.

Note

Los resúmenes de comentarios son transitorios. Si actualizas una solicitud de cambios, el resumen desaparecerá. Los resúmenes de contenido no incluyen comentarios sobre la solicitud de extracción en general, solo los comentarios que se dejan sobre las diferencias de código en las revisiones de la solicitud de extracción.

Para crear un resumen de los comentarios en una solicitud de extracción

1. Navega hasta la solicitud de extracción que creaste en el procedimiento anterior.

Tip

Si lo prefieres, puedes usar cualquier solicitud de extracción abierta en tu proyecto. En la barra de navegación, selecciona Código, selecciona Solicitudes de extracción y selecciona cualquier solicitud de extracción abierta.

2. Agrega algunos comentarios a la solicitud de extracción en Cambios si la solicitud de extracción aún no tiene comentarios.
3. En Descripción general, selecciona Crear resumen de comentarios. Cuando haya terminado, se ampliará la sección de resumen de comentarios.
4. Revisa el resumen de los comentarios dejados sobre los cambios en el código en las revisiones de la solicitud de extracción y compáralo con los comentarios de la solicitud de extracción.

Crea una incidencia y asígnala a Amazon Q

Los equipos de desarrollo crean problemas para hacer un seguimiento de su trabajo y gestionarlo, pero a veces el problema persiste porque no está claro quién debe trabajar en él, o bien porque el problema requiere investigar una parte concreta de la base de código, o bien porque hay que ocuparse primero de otras tareas urgentes. CodeCatalyst incluye la integración con un asistente de IA generativa llamado Amazon Q que puede analizar un problema en función de su título y su descripción. Si asignas el problema a Amazon Q, intentará crear un borrador de solución para que lo evalúes. Esto puede ayudarte a ti y a tu equipo a centrar y optimizar su trabajo en los problemas que requieren su atención, mientras que Amazon Q trabaja en una solución para los problemas que no tiene recursos para abordar de inmediato.

Tip

Amazon Q funciona mejor en temas simples y problemas directos. Para obtener los mejores resultados, utilice un lenguaje sencillo para explicar claramente lo que quiere que se haga.

Cuando asignes un problema a Amazon Q, CodeCatalyst marcará el problema como bloqueado hasta que confirmes cómo quieres que Amazon Q lo solucione. Requiere que respondas a tres preguntas antes de que pueda continuar:

- Si quieres confirmar cada paso que dé o si quieres que continúe sin comentarios. Si decides confirmar cada paso, puedes responder a Amazon Q con comentarios sobre el enfoque que crea para que pueda repetirlo si es necesario. Amazon Q también puede revisar los comentarios que los usuarios dejan en cualquier pull request que cree si eliges esta opción. Si decides no confirmar cada paso, Amazon Q podría completar su trabajo más rápido, pero no revisará ningún comentario que le des en el número ni en ninguna solicitud de cambios que cree.
- Si quieres permitir que actualice los archivos del flujo de trabajo como parte de su trabajo. Es posible que tu proyecto tenga flujos de trabajo configurados para iniciar ejecuciones con eventos de solicitudes de extracción. Si es así, cualquier solicitud de extracción que Amazon Q cree y que incluya la creación o actualización del flujo de trabajo YAML podría iniciar una ejecución de los flujos de trabajo incluidos en la solicitud de extracción. Como práctica recomendada, no permitas que Amazon Q trabaje en archivos de flujo de trabajo a menos que estés seguro de que no hay flujos de trabajo en tu proyecto que los ejecuten automáticamente antes de revisar y aprobar la solicitud de extracción que crea.

- En qué repositorio de origen quieres que funcione. Incluso si su proyecto tiene varios repositorios de fuentes, Amazon Q solo puede trabajar con el código de un repositorio de fuentes. No se admiten los repositorios enlazados.

Una vez que hayas hecho y confirmado tus elecciones, Amazon Q pasará el problema al estado En curso mientras intenta determinar de qué se trata la solicitud en función del título del problema y su descripción, así como del código del repositorio especificado. Creará un comentario fijo en el que proporcionará información actualizada sobre el estado de su trabajo. Tras revisar los datos, Amazon Q formulará un posible enfoque para una solución. Amazon Q registra sus acciones actualizando su comentario fijo y comentando su progreso en relación con el tema en cada etapa. A diferencia de los comentarios y respuestas anclados, no lleva un registro estrictamente cronológico de su trabajo. Más bien, coloca la información más relevante sobre su trabajo en el nivel superior del comentario anclado. Intentará crear código basándose en su enfoque y en el análisis del código que ya está en el repositorio. Si genera correctamente una posible solución, creará una rama y enviará el código a esa rama. Luego crea una solicitud de cambios que fusionará esa rama con la rama predeterminada. Cuando Amazon Q termina su trabajo, pasa el problema a En revisión para que tú y tu equipo sepan que hay código listo para que lo evalúes.

Note

Esta función solo está disponible a través de Issues. No está disponible si has configurado tu proyecto para usar Jira con la extensión de Jira Software. Además, si has personalizado el diseño de tu placa, es posible que el problema no cambie de estado. Para obtener los mejores resultados, usa esta función solo con proyectos que tengan un diseño de tablero estándar.

Una vez que hayas asignado un problema a Amazon Q, no podrás cambiar el título o la descripción del problema ni asignárselo a nadie más. Si desasignas Amazon Q del problema, finalizará su paso actual y, a continuación, dejará de funcionar. No podrá reanudar su trabajo ni reasignarse al problema una vez desasignado.

En esta parte del tutorial, crearemos tres números basados en las posibles características del código incluido en los proyectos creados con el modelo de aplicación web moderna de tres niveles: uno para añadir uno para crear una nueva criatura de myfit, otro para añadir una función de ordenación y otro para actualizar un flujo de trabajo de forma que incluya una rama con un nombre. **test**

Note

Si está trabajando en un proyecto con un código diferente, cree problemas con los títulos y las descripciones que se relacionen con esa base de código.

Para crear un problema y generar una solución para que la evalúes

1. En el panel de navegación, selecciona Problemas y asegúrate de que estás en la vista del tablero.
2. Selecciona Crear incidencia.
3. Dale al número un título que explique lo que quieres hacer en un lenguaje sencillo. Por ejemplo, para este número, introduce un título de **Create another mysfit named Quokkapus**. En Descripción, proporciona los siguientes detalles:

```
Expand the table of mysfits to 13, and give the new mysfit the following characteristics:
```

```
Name: Quokkapus
```

```
Species: Quokka-Octopus hybrid
```

```
Good/Evil: Good
```

```
Lawful/Chaotic: Chaotic
```

```
Age: 216
```

```
Description: Australia is full of amazing marsupials, but there's nothing there quite like the Quokkapus.
```

```
She's always got a friendly smile on her face, especially when she's using her eight limbs to wrap you up
```

```
in a great big hug. She exists on a diet of code bugs and caffeine. If you've got some gnarly code that needs a
```

```
assistance, adopt Quokkapus and put her to work - she'll love it! Just make sure you leave enough room for
```

```
her to grow, and keep that coffee coming.
```

4. (Opcional) Adjunta una imagen para utilizarla como miniatura y foto de perfil del mysfit al número. Si lo haces, actualiza la descripción para incluir detalles sobre qué imágenes quieres usar y por qué. Por ejemplo, puede añadir lo siguiente a la descripción: «El mysfit requiere que


los archivos de imagen se desplieguen en el sitio web. Añada estas imágenes al repositorio de origen».

 Note

En este tutorial, las imágenes adjuntas no se desplegarán en el sitio web. Puedes añadir las imágenes al sitio web tú mismo y, a continuación, dejar comentarios para que Amazon Q actualice su código para que apunte a las imágenes que quieres que utilice después de haber creado una solicitud de extracción.

Revisa la descripción y asegúrate de que contiene todos los detalles que puedan ser necesarios antes de continuar con el siguiente paso.

5. En Asignatarios, selecciona Asignar a Amazon Q.
6. En el repositorio de origen, elija el repositorio de origen que contiene el código del proyecto.
7. Desliza el botón Requerir que Amazon Q se detenga después de cada paso y espera a que se revise su selector de trabajo hasta el estado activo.

 Note

Si eliges la opción de que Amazon Q pare después de cada paso, podrás comentar el problema y tener la opción de que Amazon Q cambie su enfoque hasta tres veces en función de tus comentarios. Si eliges la opción de que Amazon Q no pare después de cada paso para poder revisar su trabajo, es posible que el trabajo avance más rápido, ya que Amazon Q no espera tus comentarios, pero no podrás influir en la dirección que tome Amazon Q dejando comentarios. Amazon Q tampoco responderá a los comentarios que se dejen en una solicitud de cambios si eliges esa opción.

8. Deje el selector Permitir que Amazon Q modifique los archivos del flujo de trabajo en estado inactivo.
9. Selecciona Crear problema. Tu vista cambia al panel de asuntos.
10. Selecciona Crear problema para crear otro problema, esta vez uno con el título **Change the get_all_myfits() API to return myfits sorted by the Age attribute**. Asigna este problema a Amazon Q y crea el problema.
11. Selecciona Crear edición para crear otra edición, esta vez con el título **Update the OnPullRequest workflow to include a branch named test in its triggers**.

Si lo desea, enlace al flujo de trabajo en la descripción. Asigne este problema a Amazon Q, pero esta vez asegúrese de que el selector Permitir que Amazon Q modifique los archivos del flujo de trabajo esté establecido en el estado activo. Crea el problema para volver al panel de problemas.

 Tip

Puede buscar archivos, incluidos los archivos de flujo de trabajo, introduciendo el símbolo at (@) e introduciendo el nombre del archivo.

Una vez que haya creado y asignado los problemas, estos pasarán a estar en curso. Amazon Q añadirá comentarios para hacer un seguimiento de su progreso dentro del tema en un comentario fijo. Si es capaz de definir un enfoque para una solución, actualizará la descripción del problema con una sección de antecedentes que contiene su análisis del código base y una sección de enfoque que detalla el enfoque propuesto para crear una solución. Si Amazon Q logra encontrar una solución al problema descrito en el problema, creará una sucursal y codificará los cambios en esa sucursal para implementar la solución propuesta. Una vez que el código esté listo, creará una solicitud de extracción para que puedas revisar los cambios de código sugeridos, añadirá un enlace a esa solicitud de extracción de información sobre el problema y pasará la edición a En revisión.

 Important

Siempre debes revisar cualquier cambio en el código de una solicitud de extracción antes de fusionarla. Los cambios de código combinados realizados por Amazon Q, como cualquier otro cambio de código, pueden afectar negativamente a la base de código y al código de infraestructura si el código fusionado no se revisa adecuadamente y contiene errores al fusionarse.

Para revisar un problema y una solicitud de cambios vinculada que contiene cambios realizados por Amazon Q

1. En Problemas, selecciona un problema asignado a Amazon Q que esté en curso. Revisa los comentarios para monitorear el progreso del bot. Si están presentes, revisa los antecedentes y acércate a los registros que figuran en la descripción del problema y, a continuación, selecciona X para cerrar el panel de problemas.
2. Ahora elige un problema asignado a Amazon Q que esté en Revisión. Revise los antecedentes y aborde los registros en la descripción del problema. Revise los comentarios para comprender las

acciones que llevó a cabo. En las solicitudes de extracción, selecciona el enlace a la solicitud de extracción junto a la etiqueta Abrir para revisar el código.

3. En la solicitud de extracción, revisa los cambios en el código. Para obtener más información, consulte [Revisión de una solicitud de extracción](#). Deja comentarios en la solicitud de cambios si quieres que Amazon Q cambie alguno de sus códigos sugeridos. Sea específico al dejar comentarios para Amazon Q para obtener mejores resultados.

Por ejemplo, al revisar la solicitud de cambios creada para **Create another mysfit named Quokkapus**, es posible que notes que hay un error tipográfico en la descripción. Puedes dejar un comentario para Amazon Q que diga: «Cambia la descripción para corregir el error tipográfico «needa» añadiendo un espacio entre «needs» y «a». Como alternativa, puedes dejar un comentario en el que se indique a Amazon Q que actualice la descripción y proporcione la descripción revisada completa para incorporarla.

Si has subido imágenes del nuevo mysfit al sitio web, puedes dejar un comentario para que Amazon Q actualice el mysfit con punteros a la imagen y una miniatura para usarlas en el nuevo mysfit.

Note

Amazon Q no responderá a los comentarios individuales. Amazon Q solo incorporará los comentarios dejados en los comentarios de las solicitudes de cambios si has elegido la opción predeterminada de dejar de hacerlo después de cada paso para su aprobación cuando creaste el problema.

4. (Opcional) Una vez que tú y otros usuarios del proyecto hayáis dejado todos los comentarios que deseáis para los cambios en el código, selecciona Crear revisión para que Amazon Q cree una revisión de la solicitud de cambios que habéis solicitado en los comentarios. Amazon Q informará del progreso en la creación de la revisión en Descripción general, no en Cambios. Asegúrate de actualizar tu navegador para ver las últimas actualizaciones de Amazon Q al crear la revisión.

Note

Solo el usuario que creó la incidencia puede crear una revisión de la solicitud de cambios. Solo puedes solicitar una revisión de una solicitud de extracción. Asegúrese

de haber abordado todos los problemas relacionados con los comentarios y de que está satisfecho con el contenido de los comentarios antes de elegir Crear revisión.

5. En este proyecto de ejemplo, se ejecuta un flujo de trabajo para cada solicitud de incorporación de cambios. Asegúrate de ver que el flujo de trabajo se ha ejecutado correctamente antes de fusionar la solicitud de extracción. También puedes optar por crear flujos de trabajo y entornos adicionales para probar el código antes de fusionarlo. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).
6. Cuando estés satisfecho con la última revisión de la solicitud de extracción, selecciona Fusionar.

Eliminar recursos

Una vez que haya completado este tutorial, considere la posibilidad de realizar las siguientes acciones para limpiar los recursos que creó durante este tutorial y que ya no necesite.

- Anule la asignación de Amazon Q de cualquier problema en el que ya no se esté trabajando. Si Amazon Q ha terminado de solucionar un problema o no ha podido encontrar una solución, asegúrate de anular la asignación de Amazon Q para evitar alcanzar la cuota máxima de funciones de IA generativa. [Para obtener más información, consulta Cómo gestionar las funciones y los precios de la IA generativa.](#)
- Mueva los problemas en los que se haya completado el trabajo a Listo.
- Si el proyecto ya no es necesario, elimínelo.

Tutorial: Creación de una aplicación completa con planos de PDK componibles

Amazon CodeCatalyst ofrece varios planos diferentes para ayudarte a empezar con tus proyectos rápidamente. Un proyecto creado con un plan incluye los recursos que necesitas, como un repositorio fuente, ejemplos de código fuente, flujos de trabajo de CI/CD, informes de creación y prueba y herramientas integradas de seguimiento de problemas. Sin embargo, a veces es posible que desee crear un proyecto de forma gradual o añadir funcionalidad a un proyecto existente creado mediante un plano. También puede hacerlo con planos. En este tutorial se muestra cómo empezar con un único plano que sienta las bases y permite almacenar todo el código del proyecto en un único repositorio. A partir de ahí, dispondrá de la flexibilidad necesaria para incorporar recursos e

infraestructuras adicionales al plan inicial, según le convenga. A través de este método básico, puede abordar requisitos específicos en varios proyectos.

En este tutorial, se muestra cómo crear varios planos del kit de desarrollo de proyectos de AWS (AWS PDK) para crear una aplicación compuesta por un sitio web de React, la API de Smithy y la infraestructura de CDK de soporte para implementarla en AWS. El PDK de AWS proporciona los componentes básicos de los patrones comunes, junto con herramientas de desarrollo para gestionar y crear sus proyectos. Para obtener más información, consulte el [repositorio de GitHub código fuente de AWS PDK](#).

Los siguientes planos de PDK están diseñados para usarse juntos a fin de crear una aplicación de forma que pueda componerse:

- [Monorepo](#): crea un proyecto a nivel raíz que gestiona las interdependencias entre los proyectos dentro del monorepo. El proyecto también proporciona almacenamiento en caché de compilación y visualización de dependencias.
- [API de Type Safe](#): crea una API que se puede definir en [Smithy](#) u [OpenAPI](#) v3, y gestiona la generación de código en tiempo de compilación para permitirle implementar e interactuar con su API de forma segura. Vende una construcción de CDK que gestiona la implementación de la API en API Gateway y configura la validación automática de las entradas.
- [Sitio web de Cloudscape React](#): crea un sitio web basado en React creado con [Cloudscape](#) que viene preintegrado con Cognito Auth y (opcionalmente) con la API que haya creado, lo que le permite llamar a su API de forma segura.
- [Infraestructura](#): crea un proyecto que configura toda la infraestructura relacionada con el CDK necesaria para implementar la aplicación. También viene preconfigurado para generar un diagrama basado en el código de CDK cada vez que se crea.
- [DevOps](#)- Crea DevOps flujos de trabajo compatibles con las construcciones que se encuentran en el kit de desarrollo de proyectos de AWS (AWS PDK).

El tutorial también incluye pasos sobre cómo ver la aplicación implementada, invitar a otros usuarios a trabajar en ella y realizar cambios en el código con solicitudes de extracción que se crean e implementan automáticamente en los recursos de la cuenta de AWS conectada cuando se fusiona la solicitud de extracción.

Cuando crea un proyecto compuesto por planos de PDK, el proyecto se crea con los siguientes recursos en un CodeCatalyst proyecto:

- Un repositorio de origen configurado como monorepo.
- Un flujo de trabajo que ejecuta el análisis de código estático y la verificación de licencias, además de compilar e implementar el código de muestra cada vez que se realiza un cambio en la rama predeterminada. Cada vez que se realizan cambios en el código, se genera un diagrama de arquitectura.
- Un panel de problemas y una lista de tareas pendientes que puede utilizar para planificar y realizar un seguimiento del trabajo.
- Un conjunto de informes de pruebas con informes automatizados.

Temas

- [Requisitos previos](#)
- [Paso 1: Crear un proyecto de monorepo](#)
- [Paso 2: Agrega la API Type Safe al proyecto](#)
- [Paso 3: Añadir un sitio web de Cloudscape React para el proyecto](#)
- [Paso 4: generar la infraestructura para implementar la aplicación en la nube de AWS](#)
- [Paso 5: Configure un DevOps flujo de trabajo para implementar su proyecto](#)
- [Paso 6: Confirma el flujo de trabajo de publicación y consulta tu sitio web](#)
- [Colabora e itera en el proyecto PDK](#)

Requisitos previos

Para crear y actualizar un proyecto, debe haber completado las tareas de la [Configuración CodeCatalyst](#) siguiente manera:

- Ten un ID de AWS constructor para iniciar sesión CodeCatalyst.
- Pertenece a un espacio y tienes asignado el rol de administrador del espacio o usuario avanzado en ese espacio. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#), [Gestión de los usuarios del espacio](#) y [Función de administrador de espacios](#).
- Tenga una cuenta de AWS asociada a su espacio y disponga del rol de IAM que creó durante el registro. Por ejemplo, durante el registro, tiene la opción de elegir crear un rol de servicio con una política de roles denominada política de roles CodeCatalystWorkflowDevelopmentRole: **SpaceName**. El rol tendrá un nombre CodeCatalystWorkflowDevelopmentRole-*spaceName* con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles,

consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#). Para conocer los pasos necesarios para crear el rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#).

Paso 1: Crear un proyecto de monorepo

Comience con el plan PDK: Monorepo para crear su base de código monorepo que sirva de base y le permita añadir planos de PDK adicionales.

Para crear un proyecto utilizando el esquema PDK - Monorepo

1. [Abra la consola en https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un proyecto.
3. En el panel de control del espacio, seleccione Crear proyecto.
4. Seleccione Empezar con un plano.
5. Elija el plano PDK - Monorepo y, a continuación, elija Siguiente.
6. En Asigne un nombre a su proyecto, introduzca el nombre que desee asignar al proyecto y los nombres de los recursos asociados. El nombre debe ser único dentro de su espacio.
7. En Recursos del proyecto, haga lo siguiente:
 - a. En Lenguaje de programación principal, elige el lenguaje en el que quieras desarrollar el código del proyecto. Puede elegir TypeScript entre Java o Python.
 - b. Elija la configuración del código
 - c. En el campo de entrada de texto del repositorio de origen, introduzca el nombre de un repositorio de origen, que creará un repositorio nuevo, o seleccione uno de los repositorios enlazados existentes. El repositorio existente debe estar vacío. Para obtener más información, consulte [Vincular un repositorio de fuentes](#).
 - d. (Opcional) En el menú desplegable Package Manager, seleccione un administrador de paquetes. Esto solo es necesario si seleccionaste TypeScript como lenguaje de programación principal.
8. (Opcional) Para obtener una vista previa del código que se generará en función de las selecciones de parámetros del proyecto que haya realizado, seleccione Ver código en Generar vista previa del proyecto.

9. (Opcional) Seleccione Ver detalles en la tarjeta del plano para ver detalles específicos del plano, como una descripción general de la arquitectura del plan, las conexiones y los permisos necesarios y el tipo de recursos que crea el plan.
10. Elija Crear proyecto para crear su proyecto monorepo. El proyecto de nivel raíz creado gestiona las interdependencias entre los proyectos dentro del monorepo, además de proporcionar almacenamiento en caché de compilación y administración de dependencias.

Para obtener más información sobre los planos de los proyectos, consulte. [Referencia del plano del proyecto](#)

El plan PDK - Monorepo solo genera la base del proyecto. Para crear una aplicación que funcione con el plano, debe añadir otros modelos de PDK, como la API Type Safe, el sitio web de Cloudscape React, la infraestructura o. DevOps En el siguiente paso, aplicará una API de Type Safe al proyecto.

Paso 2: Agrega la API Type Safe al proyecto

El plan PDK (Type Safe API) le permite definir una API mediante Smithy o OpenAPI v3. Genera paquetes de tiempo de ejecución a partir de su definición de API, que incluyen clientes para interactuar con su API y código del lado del servidor para implementar su API. El plan también genera una construcción de CDK con seguridad de tipos para cada operación de la API. Puede aplicar el modelo a un proyecto monorepo de PDK existente para añadir funciones de API al proyecto.

Para aplicar el esquema PDK: Type Safe API

1. En el panel de navegación de su proyecto de monorepo, elija Blueprints y, a continuación, seleccione Aplicar blueprint.
2. Elige el blueprint PDK: Type Safe API y, a continuación, selecciona Siguiente.
3. En Configurar el blueprint, configure los parámetros del blueprint:
 - En Idioma del modelo, elija el idioma en el que está definido el modelo de API.
 - En el campo de entrada de texto del espacio de nombres, introduce un espacio de nombres para tu API.
 - En el campo de entrada de texto del nombre de la API, introduce un nombre para la API.
 - En Idioma de la CDK, elige el idioma que prefieras para escribir la infraestructura de la CDK en la que implementar la API.

- Selecciona el menú desplegable de idiomas de los controladores y, a continuación, elige los idiomas en los que quieres implementar los controladores para las operaciones de la API.
 - Selecciona el menú desplegable Formatos de documentación y, a continuación, elige los formatos que deseas para generar la documentación de la API.
4. En la pestaña Cambios de código, revisa los cambios propuestos. La diferencia que se muestra en una solicitud de extracción muestra los cambios en tu proyecto en el momento en que se creó la solicitud de extracción.
 5. Cuando estés satisfecho con los cambios propuestos que se realizarán cuando se aplique el blueprint, selecciona Aplicar blueprint.

Una vez creada una solicitud de incorporación de cambios, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a un recurso, como archivos, utilizando el @ signo seguido del nombre del archivo.

Note

El esquema no se aplicará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

6. En la columna Estado, selecciona Solicitud de extracción pendiente para la fila del blueprint de PDK (Type Safe API) y, a continuación, selecciona el enlace de la solicitud de extracción abierta.
7. Selecciona Fusionar, elige tu estrategia de fusión preferida y, a continuación, selecciona Fusionar para incorporar los cambios del esquema aplicado.

Tras fusionar la solicitud de cambios, se generará una nueva `packages/apis/myjdkapi` carpeta en tu proyecto monorepo, que contiene todo el código fuente relacionado con la API de Type Safe configurada.

8. En el panel de navegación, selecciona Blueprints para confirmar que el estado de la API PDK (Type Safe) se muestra actualizado.

Paso 3: Añadir un sitio web de Cloudscape React para el proyecto

El plano del sitio web PDK - Cloudscape React genera un sitio web. Puedes asociar un parámetro opcional (API de Type Safe) para configurar automáticamente tu sitio web y configurar clientes de Type Safe autenticados, además de un explorador de API interactivo para probar tus distintas API.

Para aplicar el esquema PDK: Cloudscape React Website

1. En el panel de navegación de tu proyecto de monorepo, selecciona Blueprints y, a continuación, selecciona Aplicar blueprint.
2. Elige el plano del sitio web PDK - Cloudscape React y, a continuación, selecciona Siguiente.
3. En Configurar el blueprint, configure los parámetros del blueprint:
 - En el campo de entrada de texto del nombre del sitio web, introduzca un nombre para el sitio web.
 - Selecciona el menú desplegable Type Safe APIs y, a continuación, elige los esquemas de API que desees integrar en el sitio web. Al transferir una API, se configuran los clientes autenticados y se añaden las dependencias necesarias, el explorador de API y otras funciones.
4. En la pestaña Cambios de código, revisa los cambios propuestos. La diferencia que se muestra en una solicitud de extracción muestra los cambios en tu proyecto en el momento en que se creó la solicitud de extracción.
5. Cuando estés satisfecho con los cambios propuestos que se realizarán cuando se aplique el blueprint, selecciona Aplicar blueprint.

Una vez creada una solicitud de incorporación de cambios, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a un recurso, como archivos, utilizando el @ signo seguido del nombre del archivo.

Note

El esquema no se aplicará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

6. En la columna Estado, selecciona Solicitud de extracción pendiente para la fila de planos del sitio web de PDK y Cloudscape React y, a continuación, selecciona el enlace de la solicitud de extracción abierta.
7. Selecciona Fusionar, elige tu estrategia de fusión preferida y, a continuación, selecciona Fusionar para incorporar los cambios del esquema aplicado.

Tras fusionar la solicitud de cambios, se generará una nueva `packages/websites/my-website-name` carpeta dentro de tu proyecto monorepo, que contiene todo el código fuente de tu nuevo sitio web.

8. En el panel de navegación, selecciona Blueprints para confirmar que el estado del sitio web de PDK - Cloudscape React está actualizado.

A continuación, aplicará el plan PDK (Infrastructure blueprint) para generar la infraestructura necesaria para implementar su sitio web en la nube de AWS.

Paso 4: generar la infraestructura para implementar la aplicación en la nube de AWS

El plan PDK (Infrastructure Blueprint) configura un paquete que contiene todo el código de la CDK para implementar su sitio web y su API. También proporciona la generación de diagramas y la conformidad con el paquete de creación de prototipos de forma predeterminada.

Para aplicar el plan PDK: Infraestructura

1. En el panel de navegación de su proyecto monorepo, elija Blueprints y, a continuación, seleccione Aplicar blueprint.
2. Elija el PDK: plano de infraestructura y, a continuación, elija Siguiente.
3. En Configurar el blueprint, configure los parámetros del blueprint:
 - En Idioma CDK, elija el idioma con el que desee desarrollar su infraestructura.
 - En el campo de entrada de texto del nombre de la pila, introduzca el nombre de la CloudFormation pila generada para su blueprint.

Note

Toma nota del nombre de esta pila para el siguiente paso, en el que configurarás un DevOps flujo de trabajo.

- Selecciona el menú desplegable Type Safe APIs y, a continuación, elige los esquemas de API que deseas integrar en el sitio web.
 - Selecciona el menú desplegable de sitios web de Cloudscape React TS y, a continuación, selecciona los planos de sitios web que quieras implementar en tu infraestructura (por ejemplo, PDK (sitio web de Cloudscape React)).
4. En la pestaña Cambios de código, revisa los cambios propuestos. La diferencia que se muestra en una solicitud de extracción muestra los cambios en tu proyecto en el momento en que se creó la solicitud de extracción.
 5. Cuando estés satisfecho con los cambios propuestos que se realizarán cuando se aplique el blueprint, selecciona Aplicar blueprint.

Una vez creada una solicitud de incorporación de cambios, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a un recurso, como archivos, utilizando el @ signo seguido del nombre del archivo.

Note

El esquema no se aplicará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

6. En la columna Estado, selecciona Solicitud de extracción pendiente para la fila PDK (Plano de infraestructura) y, a continuación, selecciona el enlace de la solicitud de extracción abierta.
7. Selecciona Fusionar, elige tu estrategia de fusión preferida y, a continuación, selecciona Fusionar para incorporar los cambios del esquema aplicado.

Tras fusionar la solicitud de cambios, se genera una nueva `packages/infra` carpeta dentro del proyecto monorepo, que contiene la infraestructura que implementará el proyecto en la nube de AWS.

8. En el panel de navegación, selecciona Blueprints para confirmar que el estado del PDK: Infraestructura aparece actualizado.

A continuación, aplicará el PDK (DevOpsblueprint) para implementar la aplicación.

Paso 5: Configure un DevOps flujo de trabajo para implementar su proyecto

El PDK: DevOps blueprint genera los DevOps flujos de trabajo necesarios para crear e implementar su proyecto con la cuenta y el rol de AWS especificados en la configuración.

Para aplicar el PDK: blueprint DevOps

1. En el panel de navegación de su proyecto monorepo, elija Blueprints y, a continuación, seleccione Aplicar blueprint.
2. Selecciona el PDK (DevOpsblueprint) y, a continuación, selecciona Siguiente.
3. En Configurar el blueprint, configure los parámetros del blueprint:
 - Elija el SDK de Bootstrap en el entorno actual.
 - En el campo de entrada de texto del nombre de la pila, introduzca el nombre de la CloudFormation pila que desea implementar. Debe coincidir con el nombre de la pila configurado en [Paso 4: generar la infraestructura para implementar la aplicación en la nube de AWS](#) el esquema PDK: Infrastructure.
 - Elija el menú desplegable de conexión de cuentas de AWS y, a continuación, elija la cuenta de AWS que desee utilizar para los recursos. Para obtener más información, consulte [Añadir un elemento Cuenta de AWS a un espacio](#).
 - Elija el rol que utilizará para implementar su aplicación en el menú desplegable y, a continuación, elija el rol de IAM que desee usar para implementar la aplicación de su proyecto.


Note

Al crear un rol de IAM, restrinja el rol SourceArn al actual que ProjectID se encuentra en la configuración del proyecto. Para obtener más información, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName*de servicio](#).

- Selecciona el menú desplegable Región y, a continuación, elige la región en la que quieres implementar tu proyecto monorepo. La implementación solo funciona en las regiones donde existen los servicios de AWS necesarios. Para obtener más información, consulte [Servicios de AWS por región](#).
4. En la pestaña Cambios de código, revise los cambios propuestos. La diferencia que se muestra en una solicitud de extracción muestra los cambios en tu proyecto en el momento en que se creó la solicitud de extracción.

5. Cuando estés satisfecho con los cambios propuestos que se realizarán cuando se aplique el blueprint, selecciona Aplicar blueprint.

Una vez creada una solicitud de incorporación de cambios, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a un recurso, como archivos, utilizando el @ signo seguido del nombre del archivo.


 Note

El esquema no se aplicará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

6. En la columna Estado, selecciona Solicitud de extracción pendiente para la fila PDK (Plano de infraestructura) y, a continuación, selecciona el enlace de la solicitud de extracción abierta.
7. Selecciona Fusionar, elige tu estrategia de fusión preferida y, a continuación, selecciona Fusionar para incorporar los cambios del esquema aplicado.

Tras fusionar la solicitud de extracción, se generará una nueva `.codecatalyst/workflows` carpeta en tu proyecto de monorepo.

8. En el panel de navegación, selecciona Blueprints para confirmar que el estado del PDK es actualizado DevOps.

 Note

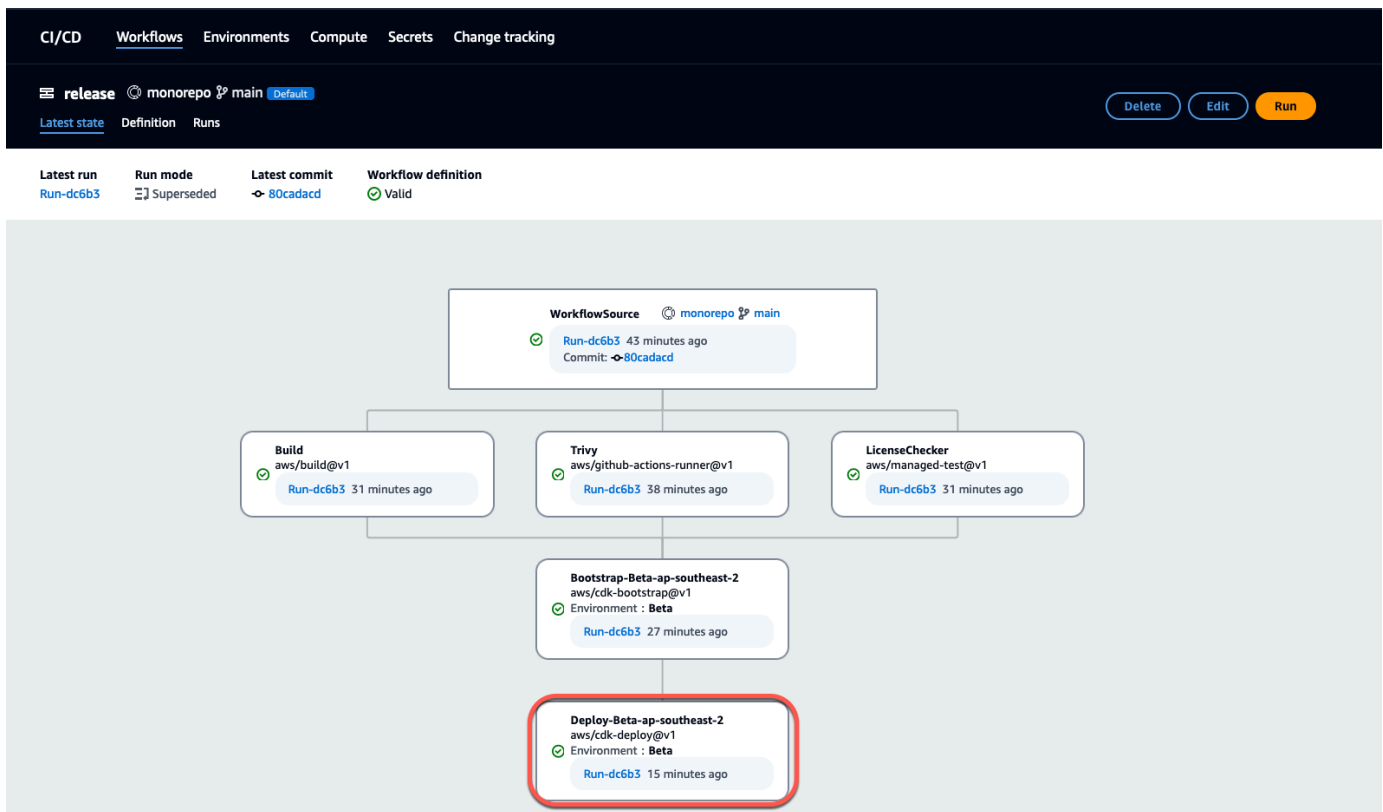
El PDK: DevOps blueprint y todos los cambios posteriores en los planos de PDK serán significativamente más lentos a partir de este momento, ya que se generan archivos de bloqueo entre bastidores para garantizar que las compilaciones e implementaciones se puedan repetir en el futuro. Generará archivos de bloqueo para todos los paquetes en cualquiera de los idiomas compatibles.

Paso 6: Confirma el flujo de trabajo de publicación y consulta tu sitio web

Una vez que hayas completado los pasos anteriores, puedes confirmar el flujo de trabajo de publicación para asegurarte de que el proyecto se está creando.

Para confirmar el flujo de trabajo de publicación y ver tu sitio web

1. En el panel de navegación de tu proyecto de monorepo, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
2. Para el flujo de trabajo de lanzamiento, selecciona la última ejecución del flujo de trabajo para ver los detalles. Para obtener más información, consulte [Ver el estado y los detalles de una sola ejecución](#).
3. Una vez que la ejecución del flujo de trabajo se haya completado correctamente, elija la última acción del flujo de trabajo (por ejemplo, Deploy-B eta-ap-southeast -2) y, a continuación, elija Variables.



4. Para ver el sitio web implementado, copie y pegue el enlace que se encuentra en la tabla de variables (por ejemplo, **myPDKAPI** websiteDistributionDomain NameXXXX) en una nueva ventana del navegador.

Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds


Logs

Summary

Configuration

Variables

Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityUserPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

Necesitas una cuenta de Amazon Cognito para iniciar sesión en tu sitio web. De forma predeterminada, el grupo de usuarios no está configurado para permitir el registro automático.

- a. Navegue hasta la [consola de AWS Cognito](#).
- b. *En la tabla **Grupos de usuarios**, elija el nombre del grupo de usuarios que coincida con el grupo de usuarios creado por el PDK: **DevOps** blueprint, que se encuentra en la tabla de **variables** (por ejemplo, **infra calculate calculate XXXXX**). `betaUserIdentityinfra` `betaUserIdentity` `IdentityPoolId` Para obtener más información, consulte [Introducción](#) a los grupos de usuarios.*

Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds

Logs

Summary

Configuration

Variables

Output variables (7)

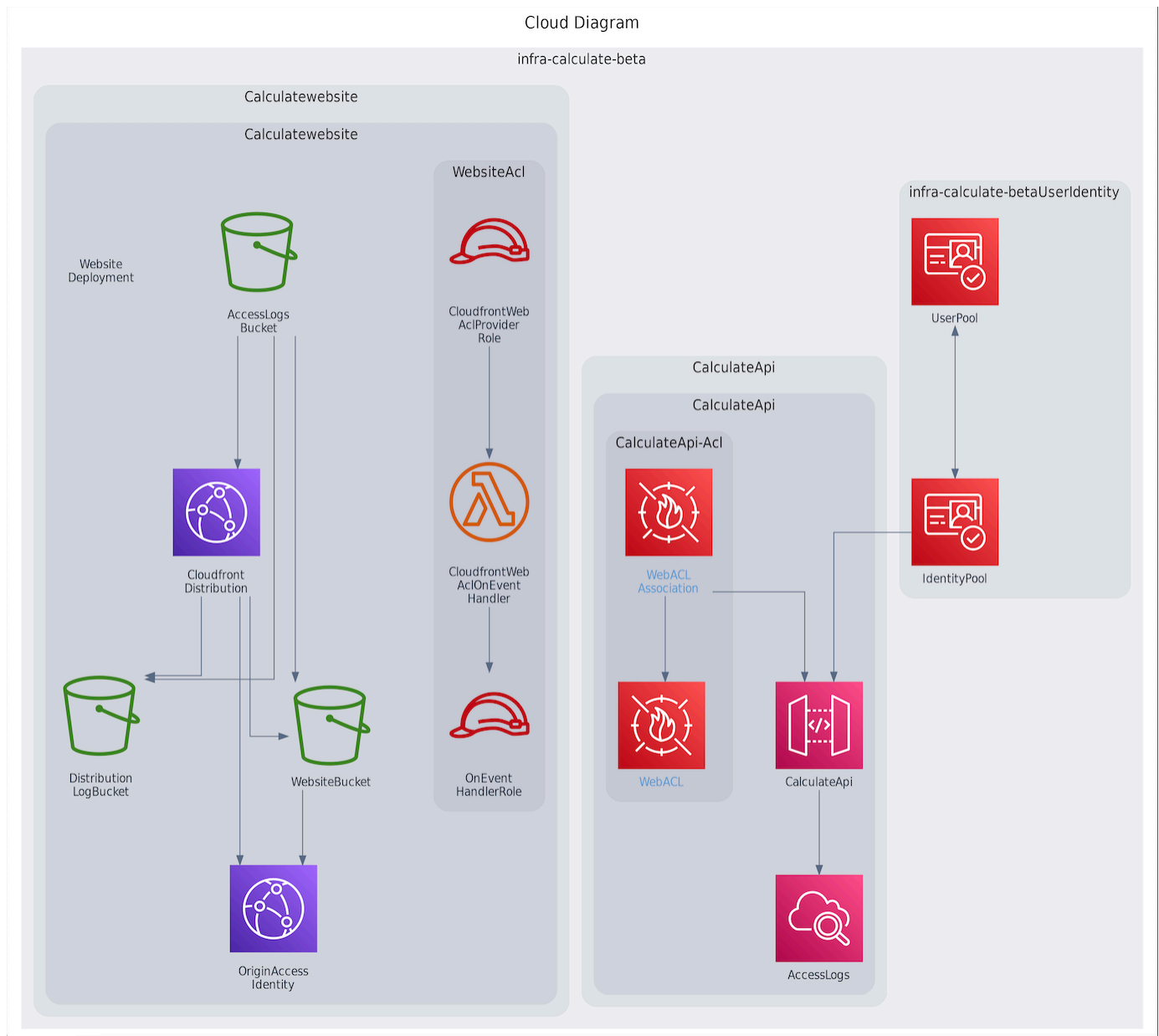
Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

- c. Seleccione la opción Crear un usuario.
 - d. Configure los parámetros de información del usuario:
 - En Mensaje de invitación, selecciona Enviar una invitación por correo electrónico.
 - En el campo de entrada de texto Nombre de usuario, introduce un nombre de usuario.
 - En el campo de entrada de texto Dirección de correo electrónico, introduzca un nombre de usuario.
 - En Contraseña temporal, selecciona Generar una contraseña.
 - e. Seleccione la opción Crear un usuario.
 - f. Navegue hasta la cuenta de correo electrónico que ingresó para los parámetros de información del usuario y abra un correo electrónico con una contraseña temporal. Tome nota de la contraseña.
 - g. Vuelva al sitio web desplegado, introduzca el nombre de usuario que creó y la contraseña temporal que recibió y, a continuación, seleccione Iniciar sesión.
5. (Opcional) Una vez que la ejecución del flujo de trabajo se haya completado correctamente, también podrá ver el diagrama generado. Seleccione la pestaña Artefactos CodeCatalyst, elija Descargar para la fila del diagrama y, a continuación, abra los archivos descargados.

The screenshot shows the Amazon CodeCatalyst interface. At the top, there are navigation tabs: CI/CD, Workflows, Environments, Compute, Secrets, and Change tracking. Below this, the workflow name 'Run-ef953' is displayed, along with buttons for 'Edit workflow' and 'New run'. The 'Artifacts' tab is selected, showing a table of artifacts. The 'Diagram' artifact is highlighted with a red circle.

Status	Run mode	Trigger	Start time	Duration	End time
✔ Succeeded	Superseded	Started by a9865766	about 14 hours ago	25 minutes 1 second	about 14 hours ago

Artifact name	Files	Produced by	Consumed by
Built	Download	Build	Bootstrap-Beta-ap-southeast-2 Deploy-Beta-ap-southeast-2
Diagram	Download	Build	-
bdd254b65baac169f6ac50e8175ce6d930c1fcb086dec59808d3e0170ae2291d_report	Download	Build	-
a093422585a8a4cb763d89a0fa8e76744a80830fe24724c7e7943a50ec479240_report	Download	Trivy	-



Colabora e itera en el proyecto PDK

Una vez configurado el proyecto, puede realizar cambios en el código fuente. También puedes invitar a otros miembros del espacio a trabajar en el proyecto. Los planos de PDK le permiten crear su aplicación de forma iterativa, añadiendo únicamente lo que necesite, cuando lo necesite, sin perder el control total de la configuración de cada plano.

Temas

- [Paso 1: Invita a los miembros a tu proyecto](#)

- [Paso 2: Crea temas para colaborar y realizar un seguimiento del trabajo](#)
- [Paso 3: Visualiza tu repositorio de código fuente](#)
- [Paso 4: Crea un entorno de desarrollo y realiza cambios en el código](#)
- [Paso 5: Inserta y fusiona los cambios de código](#)

Paso 1: Invita a los miembros a tu proyecto

Puedes usar la consola para invitar a usuarios a tu proyecto. Puedes invitar a miembros de tu espacio o añadir nombres ajenos a tu espacio.

Para invitar a usuarios a tu proyecto, debes iniciar sesión con el rol de administrador del proyecto o administrador del espacio.

No es necesario que invite a un usuario con el rol de administrador del espacio a su proyecto, ya que este ya tiene acceso implícito a todos los proyectos del espacio.

Al invitar a un usuario a su proyecto (sin asignar la función de administrador del espacio), el usuario aparecerá en la tabla de miembros del proyecto, en los proyectos, y en la tabla de miembros del proyecto, en los espacios.

Para invitar a un miembro a tu proyecto desde la pestaña de configuración del proyecto

1. Vaya a su proyecto.

Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.

2. En el panel de navegación, selecciona Configuración del proyecto.
3. Seleccione la pestaña Miembros.
4. En Miembros del proyecto, elija Invitar a un nuevo miembro.
5. Escribe la dirección de correo electrónico del nuevo miembro, elige el rol de este miembro y, a continuación, selecciona Invitar. Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Para invitar a un miembro a tu proyecto desde la página de resumen del proyecto

1. Vaya a su proyecto.

 Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.


2. Pulse el botón Members +.
3. Escriba la dirección de correo electrónico del nuevo miembro, elija el rol de este miembro y, a continuación, elija Invitar. Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Paso 2: Crea temas para colaborar y realizar un seguimiento del trabajo

CodeCatalyst te ayuda a realizar un seguimiento de las funciones, las tareas, los errores y cualquier otro trabajo relacionado con tu proyecto que presente problemas. Puedes crear problemas para hacer un seguimiento del trabajo y las ideas necesarias. De forma predeterminada, cuando creas una incidencia, se añade a tu cartera de trabajo pendiente. Puedes mover las cuestiones a un tablero donde puedes hacer un seguimiento del trabajo en curso. También puedes asignar una incidencia a un miembro específico del proyecto. En este paso, cree una incidencia para realizar cambios en su proyecto de PDK.

Para crear una edición


1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto de monorepo en el que desee crear una edición.
3. En la página de inicio del proyecto, selecciona Crear edición. Como alternativa, en el panel de navegación, selecciona Problemas.
4. Selecciona Crear problema.

 Note

También puedes añadir las incidencias en línea cuando utilices una vista de cuadrícula.

5. Introduce un título para el número.
6. (Opcional) Introduce una descripción. Para este problema, introduzca la siguiente descripción: a change in the `src/mysfit_data.json` file.. Puede usar Markdown para agregar formato.
7. (Opcional) Elige un estado, una prioridad o una estimación para el problema.

8. (Opcional) Añada una etiqueta existente o cree una nueva y agréguela seleccionando + Añadir etiqueta.
 - a. Para añadir una etiqueta existente, selecciónela de la lista. Puede introducir un término de búsqueda en el campo para buscar en el proyecto todas las etiquetas que contengan ese término.
 - b. Para crear una etiqueta nueva y añadirla, introduce el nombre de la etiqueta que deseas crear en el campo de búsqueda y pulsa enter.
9. (Opcional) Agregue un cesionario seleccionando + Agregar un cesionario. Puedes añadirte rápidamente como cesionario seleccionando + Añadirme.

 Tip

Puedes asignar un problema a Amazon Q para que Amazon Q intente resolverlo. Para obtener más información, consulte [Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo](#).

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas para el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

10. (Opcional) Añada un campo personalizado existente o cree uno nuevo. Los problemas pueden tener varios campos personalizados.
 - a. Para añadir un campo personalizado existente, elija el campo personalizado de la lista. Puede introducir un término de búsqueda en el campo para buscar en todos los campos personalizados que contengan ese término en el proyecto.
 - b. Para crear un campo personalizado nuevo y añadirlo, introduzca el nombre del campo personalizado que desee crear en el campo de búsqueda y pulse Entrar. A continuación, elija el tipo de campo personalizado que desee crear y establezca un valor.
11. Selecciona Crear incidencia. Aparece una notificación en la esquina inferior derecha: si el problema se creó correctamente, aparecerá un mensaje de confirmación que indica que el problema se creó correctamente. Si el problema no se ha creado correctamente, aparece un mensaje de error con el motivo del error. A continuación, puede seleccionar Reintentar para editar y volver a intentar crear el problema, o bien seleccionar Descartar para descartar el problema. Ambas opciones descartarán la notificación.

Note

No puedes vincular una solicitud de cambios a una publicación al crearla. Sin embargo, puedes [editarla](#) después de crearla para añadir enlaces a las solicitudes de cambios.

Para obtener más información, consulte [Problemas en CodeCatalyst](#).

Paso 3: Visualiza tu repositorio de código fuente

Puedes ver los repositorios de origen asociados a un proyecto en Amazon CodeCatalyst. En el caso de los repositorios de origen CodeCatalyst, la página de información general de un repositorio proporciona un resumen rápido de la información y la actividad de ese repositorio, que incluye:

- La descripción del repositorio, si la hay
- El número de sucursales del repositorio
- El número de solicitudes de cambios abiertas para el repositorio
- El número de flujos de trabajo relacionados con el repositorio
- Los archivos y carpetas de la rama predeterminada o de la rama que elija
- El título, el autor y la fecha de la última confirmación en la rama mostrada
- El contenido del archivo README.md renderizado en Markdown, si se incluye algún archivo README.md

Esta página también proporciona enlaces a las confirmaciones, ramificaciones y solicitudes de extracción del repositorio, así como una forma rápida de abrir, ver y editar archivos individuales.

Note

No puedes ver esta información sobre los repositorios enlazados en la CodeCatalyst consola. Para ver información sobre los repositorios enlazados, selecciona el enlace de la lista de repositorios para abrir ese repositorio en el servicio que lo aloja.

Para navegar a los repositorios de origen de un proyecto

1. Navegue hasta su proyecto y realice una de las siguientes acciones:

- En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio.
 - En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen. En Repositorios de origen, elige el nombre del repositorio de la lista. Puede filtrar la lista de repositorios escribiendo parte del nombre del repositorio en la barra de filtros.
2. En la página de inicio del repositorio, consulta el contenido del repositorio y la información sobre los recursos asociados, como el número de solicitudes de incorporación de datos y los flujos de trabajo. De forma predeterminada, se muestra el contenido de la rama predeterminada. Puede cambiar la vista seleccionando una rama diferente de la lista desplegable.

 Tip

También puedes navegar rápidamente a los repositorios de tu proyecto seleccionando Ver el código del proyecto en la página de resumen del proyecto.

Paso 4: Crea un entorno de desarrollo y realiza cambios en el código

En este paso, crea un entorno de desarrollo y realiza cambios en el código que luego se fusionan en la rama principal. Si bien este tutorial le guía a través de un sencillo proyecto de AWS PDK, también puede seguir un ejemplo más complejo que se proporciona en el repositorio de [AWS PDK GitHub](#).

Para crear un entorno de desarrollo con una nueva rama

1. En el panel de navegación de tu proyecto monorepo, realiza una de las siguientes acciones:
 - Elija Descripción general y, a continuación, vaya a la sección Mis entornos de desarrollo.
 - Selecciona Código y, a continuación, selecciona Entornos de desarrollo.
 - Elija Código, repositorios de código fuente y, a continuación, elija el repositorio monorepo para el que desee crear un entorno de desarrollo.
2. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
3. Seleccione Clonar un repositorio.
4. Elija el repositorio que desea clonar, elija Trabajar en una nueva ramificación, introduzca el nombre de la ramificación en el campo Nombre de ramificación, y elija una ramificación a partir de la cual crear la ramificación nueva en el menú desplegable Crear ramificación desde.

Note

Si creas un entorno de desarrollo desde la página de repositorios de código fuente o desde un repositorio de código fuente específico, no necesitas elegir un repositorio. El entorno de desarrollo se creará a partir del repositorio de origen que hayas elegido en la página de repositorios de origen.

5. (Opcional) En Alias: opcional, introduce un alias para el entorno de desarrollo.
6. (Opcional) Pulse el botón de edición de la configuración del entorno de desarrollo para editar la configuración de procesamiento, almacenamiento o tiempo de espera del entorno de desarrollo.
7. (Opcional) En Amazon Virtual Private Cloud (Amazon VPC): opcional, selecciona una conexión de VPC que quieras asociar a tu entorno de desarrollo en el menú desplegable.

Si se establece una VPC predeterminada para su espacio, sus entornos de desarrollo se ejecutarán conectados a esa VPC. Puede anular esto asociando una conexión de VPC diferente. Además, tenga en cuenta que los entornos de desarrollo conectados a VPC no son compatibles con AWS Toolkit.

Note

Al crear un entorno de desarrollo con una conexión de VPC, se crea una nueva interfaz de red dentro de la VPC. CodeCatalyst interactúa con esta interfaz mediante el rol de VPC asociado. Además, asegúrese de que el bloque CIDR de IPv4 no esté configurado para el 172.16.0.0/12 rango de direcciones IP.

8. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo.

Una vez que se ejecute su entorno de desarrollo, puede trabajar con la aplicación de muestra generada CodeCatalyst realizando cambios en el código con solicitudes de extracción que se crean e implementan automáticamente en los recursos de la cuenta de AWS conectada cuando se fusiona la solicitud de extracción. El monorepo vende un archivo de desarrollo para que todas las dependencias globales y los tiempos de ejecución necesarios estén presentes automáticamente.

Para cambiar el código de tu proyecto

1. En una terminal funcional de tu entorno de desarrollo, navega hasta tu proyecto monorepo y, a continuación, instala las dependencias del proyecto ejecutando el siguiente comando:

```
npx projen install
```

2. Navega hasta `packages/apis/myjdkapi/model/src/main/smithy/operations/say-hello.smithy`, que define un ejemplo de operación de API. En este tutorial, crearás una `Calculate` operación sencilla que suma dos números. Realice un cambio en el código para definir esta operación, incluidas sus entradas y salidas.

Ejemplo:

```
$version: "2"
namespace com.aws

@http(method: "POST", uri: "/calculate")
@handler(language: "typescript")
operation Calculate {
  input := {
    @required
    numberA: Integer
    @required
    numberB: Integer
  }
  output := {
    @required
    result: Integer
  }
}
```

El `@handler` rasgo indica a la API Type Safe que implementará esta operación como un controlador de AWS Lambda escrito en él. TypeScript La API Type Safe generará un código auxiliar para esta operación que podrá implementar en él. TypeScript Se añade la `@required` característica, lo que significa que la pasarela de API que se despliegue la aplicará en tiempo de ejecución. Para obtener más información, consulta la [documentación de Smithy](#).

3. Cambie el nombre del `/say-hello.smithy` archivo por uno que se ajuste a los cambios de código (por ejemplo, `calculate.smithy`)
4. Navegue hasta el `packages/apis/myjdkapi/model/src/main/smithy/main.smithy` código y realice un cambio en el mismo para conectar la operación. Puede mostrar la

Calculate operación definida en el `/calculate.smithy` listándola en el `operations` campo de este archivo.

Ejemplo:

```
$version: "2"
namespace com.aws

use aws.protocols#restJson1

/// A sample smithy api
@restJson1
service MyPDKApi {
  version: "1.0"
  operations: [Calculate]
  errors: [
    BadRequestError
    NotAuthorizedError
    InternalFailureError
  ]
}
```

5. Cree los cambios ejecutando el siguiente comando:

```
npx projen build
```

Note

Si lo desea, puede pasar una `--parallel X` bandera, que distribuirá la compilación entre los X núcleos.

Como se agregó la `@handler` característica, se generan los siguientes archivos una vez completada la compilación:

- `/packages/apis/myjdkapi/handlers/typescript/src/calculate.ts`
- `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts`

6. Navega hasta `packages/apis/myjdkapi/handlers/typescript/src/calculate.ts` el código y realiza los cambios en él. Este archivo es el controlador del servidor que se invoca para la API.

```

import {
  calculateHandler,
  CalculateChainedHandlerFunction,
  INTERCEPTORS,
  Response,
  LoggingInterceptor,
} from 'mypdkapi-typescript-runtime';

/**
 * Type-safe handler for the Calculate operation
 */
export const calculate: CalculateChainedHandlerFunction = async (request) => {
  LoggingInterceptor.getLogger(request).info('Start Calculate Operation');

  const { input } = request;

  return Response.success({
    result: input.body.numberA + input.body.numberB,
  });
};

/**
 * Entry point for the AWS Lambda handler for the Calculate operation.
 * The calculateHandler method wraps the type-safe handler and manages marshalling
 * inputs and outputs
 */
export const handler = calculateHandler(...INTERCEPTORS, calculate);

```

7. Navega hasta el `/packages/apis/mypdkapi/handlers/typescript/test/calculate.test.ts` archivo y realiza cambios en el código para actualizar las pruebas unitarias.

Ejemplo:

```

import {
  CalculateChainedRequestInput,
  CalculateResponseContent,
} from 'mypdkapi-typescript-runtime';
import {
  calculate,
} from '../src/calculate';

```

```
// Common request arguments
const requestArguments = {
  chain: undefined as never,
  event: {} as any,
  context: {} as any,
  interceptorContext: {
    logger: {
      info: jest.fn(),
    },
  },
} satisfies Omit<CalculateChainedRequestInput, 'input'>;

describe('Calculate', () => {

  it('should return correct sum', async () => {
    const response = await calculate({
      ...requestArguments,
      input: {
        requestParameters: {},
        body: {
          numberA: 1,
          numberB: 2
        }
      },
    });

    expect(response.statusCode).toBe(200);
    expect((response.body as CalculateResponseContent).result).toEqual(3);
  });
});
```

8. Navegue hasta el `/packages/infra/main/src/constructs/apis/myjdkapi.ts` archivo y realice cambios en el código para añadir una integración a la Calculate operación en su infraestructura de CDK. La construcción de la API tiene una propiedad de integraciones, en la que puede transferir la implementación que agregó anteriormente. Como estás usando la `@handler` característica en tu modelo de Smithy para la Calculate operación, puedes usar la construcción `CalculateFunction` CDK generada, que está preconfigurada, para apuntar a la implementación de tu controlador.

Ejemplo:


```
import { UserIdentity } from "@aws/pdk/identity";
import { Authorizers, Integrations } from "@aws/pdk/type-safe-api";
import { Stack } from "aws-cdk-lib";
import { Cors } from "aws-cdk-lib/aws-apigateway";
import {
  AccountPrincipal,
  AnyPrincipal,
  Effect,
  PolicyDocument,
  PolicyStatement,
} from "aws-cdk-lib/aws-iam";
import { Construct } from "constructs";
import { Api, CalculateFunction } from "calculateapi-typescript-infra";

/**
 * Api construct props.
 */
export interface CalculateApiProps {
  /**
   * Instance of the UserIdentity.
   */
  readonly userIdentity: UserIdentity;
}

/**
 * Infrastructure construct to deploy a Type Safe API.
 */
export class CalculateApi extends Construct {
  /**
   * API instance
   */
  public readonly api: Api;

  constructor(scope: Construct, id: string, props?: CalculateApiProps) {
    super(scope, id);

    this.api = new Api(this, id, {
      defaultAuthorizer: Authorizers.iam(),
      corsOptions: {
        allowOrigins: Cors.ALL_ORIGINS,
        allowMethods: Cors.ALL_METHODS,
      },
      integrations: {
```

```

    calculate: {
      integration: Integrations.lambda(new CalculateFunction(this,
"CalculateFunction"))
    }
  },
  policy: new PolicyDocument({
    statements: [
      // Here we grant any AWS credentials from the account that the prototype
is deployed in to call the api.
      // Machine to machine fine-grained access can be defined here using more
specific principals (eg roles or
      // users) and resources (ie which api paths may be invoked by which
principal) if required.
      // If doing so, the cognito identity pool authenticated role must still
be granted access for cognito users to
      // still be granted access to the API.
      new PolicyStatement({
        effect: Effect.ALLOW,
        principals: [new AccountPrincipal(Stack.of(this).account)],
        actions: ["execute-api:Invoke"],
        resources: ["execute-api:/*"],
      }),
      // Open up OPTIONS to allow browsers to make unauthenticated preflight
requests
      new PolicyStatement({
        effect: Effect.ALLOW,
        principals: [new AnyPrincipal()],
        actions: ["execute-api:Invoke"],
        resources: ["execute-api:/*/OPTIONS/*"],
      }),
    ],
  }),
});

// Grant authenticated users access to invoke the api
props?.userIdentity.identityPool.authenticatedRole.addToPrincipalPolicy(
  new PolicyStatement({
    effect: Effect.ALLOW,
    actions: ["execute-api:Invoke"],
    resources: [this.api.api.arnForExecuteApi("", "/*", "")],
  }),
);
}

```

```
}
```

9. Cree los cambios ejecutando el siguiente comando:

```
npx projen build
```

Una vez que el proyecto haya terminado de crearse, podrá ver el diagrama generado actualizado, que se encuentra en `el/packages/infra/main/cdk.out/cdkgraph/diagram.png`. El diagrama muestra cómo se agrega la función y cómo se conecta a la API creada. A medida que se modifica el código CDK, este diagrama también se actualiza.

Ahora puedes implementar tus cambios insertándolos y fusionándolos en la rama principal de tu repositorio.

Paso 5: Inserta y fusiona los cambios de código

Confirma e inserta los cambios en el código, que luego podrás fusionarlos en la rama principal de tu repositorio de código fuente.

Para insertar los cambios en tu rama de funciones

- Ejecuta los siguientes comandos para confirmar e insertar los cambios en tu rama de funciones:

```
git add .
```

```
git commit -m "my commit message"
```

```
git push
```

Al presionar los cambios, se activa un nuevo flujo de trabajo para tu rama de funciones, que puedes ver en la CodeCatalyst consola. A continuación, puedes crear una solicitud de cambios para combinar los cambios con la rama principal de tu repositorio de origen. Al fusionar la rama de funciones con la rama principal, se activa el flujo de trabajo de publicación. También puedes vincular la solicitud de cambios a tu edición.

Para crear una solicitud de cambios y vincularla a tu publicación

1. En tu proyecto de monorepo, realiza una de las siguientes acciones:

- En el panel de navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, selecciona Crear solicitud de extracción.
 - En la página de inicio del repositorio, selecciona Más y, a continuación, selecciona Crear solicitud de extracción.
 - En la página del proyecto, selecciona Crear solicitud de extracción.
2. En el repositorio de fuentes, asegúrate de que el repositorio de fuentes especificado es el que contiene el código confirmado. Esta opción solo aparece si no creaste la solicitud de extracción desde la página principal del repositorio.
 3. En la rama de destino, elige la rama principal en la que deseas combinar el código después de revisarlo.
 4. En la rama Fuente, elige la rama de funciones que contiene el código confirmado.
 5. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios a entender qué es lo que hay que revisar y por qué.
 6. (Opcional) En la descripción de la solicitud de retirada, proporciona información como un enlace a los problemas o una descripción de tus cambios.

 Tip

Puedes elegir Escribir una descripción para que yo genere CodeCatalyst automáticamente una descripción de los cambios incluidos en la solicitud de cambios. Puedes realizar cambios en la descripción generada automáticamente después de añadirla a la solicitud de extracción.

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulta [Cómo gestionar las funciones de IA generativa en Amazon CodeCatalyst](#).

7. En Problemas, selecciona Vincular problemas y, a continuación, elige el problema en [Paso 2: Crea temas para colaborar y realizar un seguimiento del trabajo](#) el que lo creaste. Para desvincular un problema, selecciona el icono de desvincular.
8. (Opcional) En Revisores obligatorios, selecciona Agregar revisores obligatorios. Elija de la lista de miembros del proyecto para añadirlos. Los revisores necesarios deben aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.

 Note

No puedes añadir un revisor como revisor obligatorio y como revisor opcional. No puedes añadirte a ti mismo como revisor.

9. (Opcional) En Revisores opcionales, selecciona Añadir revisores opcionales. Elija de la lista de miembros del proyecto para añadirlos. Los revisores opcionales no tienen que aprobar los cambios como requisito para poder fusionar la solicitud de extracción en la sucursal de destino.
10. Los revisores o tú mismo deberán revisar y fusionar tu solicitud de incorporación a la sucursal principal. Para obtener más información, consulte [Fusión de una solicitud de extracción](#).

Cuando tus cambios se fusionan en la rama principal de tu repositorio de origen, se activa automáticamente un nuevo flujo de trabajo.

11. Una vez finalizada la fusión, puedes mover la edición a Listo.
 - a. En el panel de navegación, selecciona Problemas.
 - b. Selecciona el problema creado en [Paso 2: Crea temas para colaborar y realizar un seguimiento del trabajo](#), selecciona el menú desplegable Estado y, a continuación, selecciona Listo.

El flujo de trabajo de lanzamiento despliega la aplicación después de ejecutarla correctamente, para que puedas ver los cambios.

Para confirmar el flujo de trabajo de lanzamiento y ver tu sitio web

1. En el panel de navegación de tu proyecto de monorepo, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
2. Para el flujo de trabajo de lanzamiento, selecciona la última ejecución del flujo de trabajo para ver los detalles. Para obtener más información, consulte [Ver el estado y los detalles de una sola ejecución](#).
3. Una vez que la ejecución del flujo de trabajo se haya completado correctamente, elija la última acción del flujo de trabajo (Deploy-B eta-ap-souteast -2) y, a continuación, elija Variables.
4. Para ver el sitio web implementado, copie y pegue el enlace de la fila **myPDKAPI** websiteDistributionDomain NameXXXXX en una nueva ventana del navegador.
5. Introduzca el nombre de usuario y la contraseña que creó y, a continuación, seleccione Iniciar sesión [Paso 6: Confirma el flujo de trabajo de publicación y consulta tu sitio web](#).

6. (Opcional) Pruebe los cambios en su aplicación.
 - a. Selecciona el menú desplegable POST.
 - b. Introduzca dos valores para `numberA` y `number B`, a continuación, seleccione Ejecutar.
 - c. Confirme los resultados en el cuerpo de la respuesta.

Con el tiempo, las versiones del catálogo de los planos del PDK pueden cambiar. Puede cambiar los planos de su proyecto por las versiones del catálogo para mantenerse al día con los cambios más recientes. Puede ver los cambios en el código y los entornos afectados antes de cambiar las versiones del plan de su proyecto. Para obtener más información, consulte [Actualización de un plano en un proyecto](#).

Espacios en CodeCatalyst

Creas un espacio que te represente a ti, a tu empresa, departamento o grupo, y proporciona un lugar en el que tus equipos de desarrollo pueden gestionar los proyectos. Debes crear un espacio para añadir proyectos, miembros y los recursos de nube asociados que crees en Amazon CodeCatalyst.

Note

Los nombres de los espacios deben ser únicos en todos los espacios CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados.

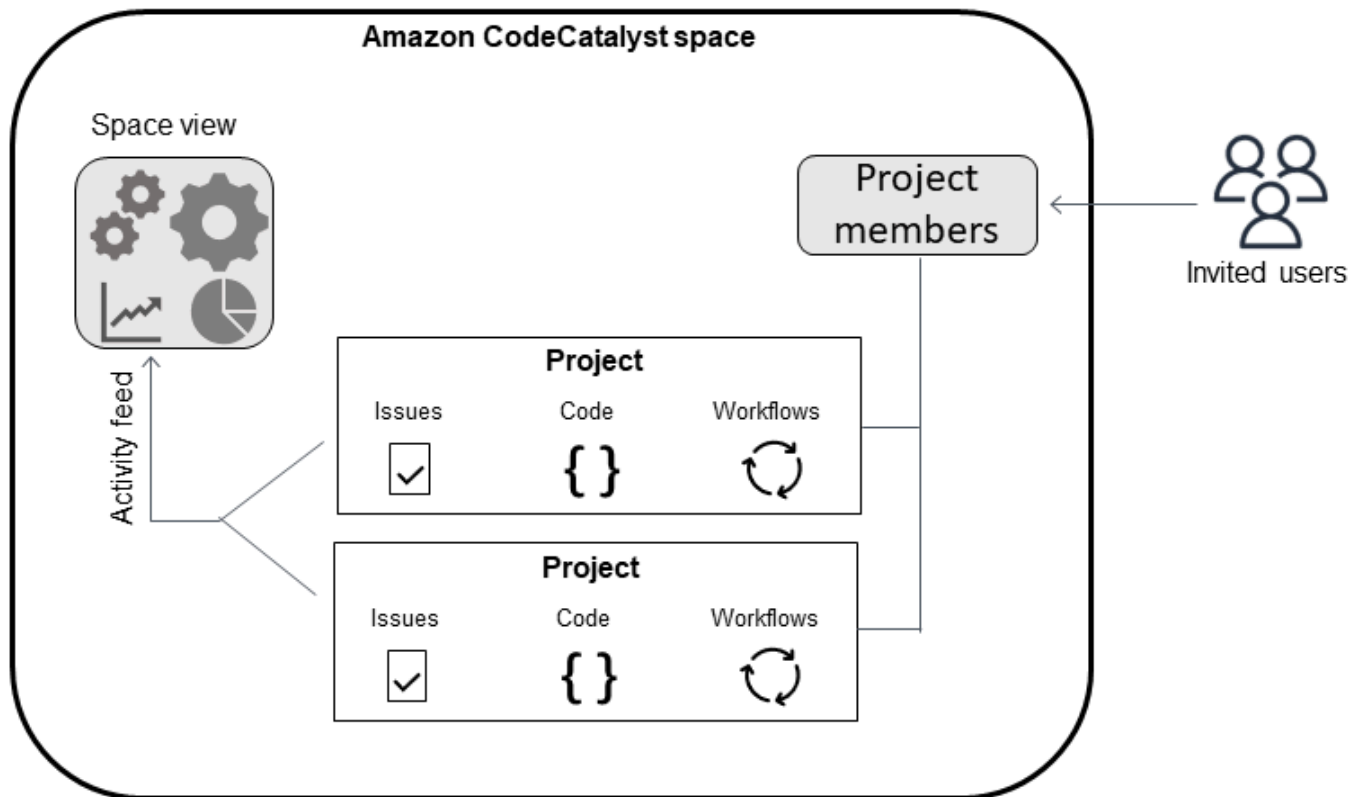
Al crear un espacio, se le asigna automáticamente la función de administrador del espacio. Puede añadir este rol a otros usuarios del espacio.

Con la función de administrador del espacio, puede administrar el espacio de la siguiente manera:

- Agregue otros administradores de espacio al espacio
- Cambie los roles y permisos de los miembros
- Edita o elimina el espacio
- Crea proyectos e invita a miembros al proyecto
- Vea una lista de todos los proyectos del espacio
- Consulta el feed de actividades de todos los proyectos del espacio

Al crear un espacio, se le añade automáticamente al espacio con dos funciones: la función de administrador del espacio y la función de administrador del proyecto que creó como parte de la creación del espacio. Los usuarios adicionales se añaden automáticamente como miembros al espacio cuando aceptan invitaciones a proyectos. Esta pertenencia al espacio no otorga ningún permiso en el espacio. Lo que los usuarios pueden hacer en un espacio viene determinado por la función que desempeñe el usuario en un proyecto específico.

Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).



Las siguientes son consideraciones adicionales para añadir cuentas:

- Hay un one-to-one mapeo de la conexión de la cuenta Cuenta de AWS a un espacio. Se Cuenta de AWS puede añadir una sola a varios espacios diferentes. Las cuentas de AWS para las que implemente no necesitan ser únicas y pueden usarse en más de un espacio.
- Cuentas de AWS las que se agreguen a un CodeCatalyst espacio se pueden usar en cualquier proyecto de ese espacio.
- Si bien cada entorno puede admitir varios Cuentas de AWS, solo puedes usar una cuenta por entorno en una acción.
- La facturación se configura a nivel de espacio. Se pueden configurar varias cuentas para la facturación, pero solo una puede estar activa en un CodeCatalyst espacio. Solo se Cuenta de AWS puede usar una como cuenta de facturación para un espacio CodeCatalyst. Si una cuenta ya está en uso para un espacio, debes usar una cuenta de facturación diferente para el espacio adicional.
- Tras crear una conexión, debe añadir funciones de AWS IAM a la conexión si el flujo de trabajo debe acceder a esas funciones de IAM con su CodeCatalyst entorno. Para obtener más información sobre cómo se utilizan los entornos, consulte. [Trabajo con entornos](#)

Temas

- [Crear un espacio que admita a los usuarios de AWS Builder ID](#)
- [Edición de un espacio](#)
- [Eliminar un espacio en CodeCatalyst](#)
- [Supervisión de la actividad de un espacio](#)
- [Cuentas de AWS Administrar un espacio](#)
- [Gestión de las funciones de IAM para las cuentas conectadas](#)
- [Gestión de los usuarios del espacio](#)
- [Gestión de equipos](#)
- [Gestión de los recursos de la máquina](#)
- [Administración de entornos de desarrollo para un espacio](#)
- [Cuotas de espacios en CodeCatalyst](#)

Crear un espacio que admita a los usuarios de AWS Builder ID

La primera vez que te registres en Amazon CodeCatalyst con tu ID de AWS constructor, tendrás que crear un espacio. Para obtener más información, consulte [Configuración CodeCatalyst](#). Puede optar por crear espacios adicionales para satisfacer las necesidades de su empresa.

Note

Los nombres de los espacios deben ser únicos en todos ellos CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados.

La información de esta guía se proporciona para crear espacios CodeCatalyst que admitan a los usuarios de AWS Builder ID. Los pasos para configurar y administrar un espacio que admita la federación de identidades se indican en la Guía CodeCatalyst del administrador. Para trabajar con espacios configurados para la federación de identidades, consulte [Configuración y administración de CodeCatalyst espacios](#) en la Guía del CodeCatalyst administrador de Amazon.

Para crear espacios adicionales que admitan a los usuarios de AWS Builder ID, debes tener asignado el rol de administrador de espacios.

Note

Al crear un espacio adicional, no se le pide que cree un proyecto. Para obtener información sobre cómo crear proyectos en un espacio, consulte [Crear un proyecto en Amazon CodeCatalyst](#).

Para crear otro espacio

1. En AWS Management Console, asegúrese de haber iniciado sesión con la misma Cuenta de AWS que desea asociar a su CodeCatalyst espacio.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

4. Selecciona Crear espacio.
5. En la página Crear un espacio, en Nombre del espacio, introduzca un nombre para el espacio. No podrá cambiarlo más adelante.

Note

Los nombres de los espacios deben ser únicos en todos ellos CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados.

6. En Región de AWS, elija la región en la que desee almacenar los datos del espacio y del proyecto. No podrá cambiarlo más adelante.
7. En Cuenta de AWS ID, introduce el identificador de doce dígitos de la cuenta que quieres conectar a tu espacio.

En el token de verificación de la AWS cuenta, copia el ID del token generado. El token se copia automáticamente, pero es posible que desee almacenarlo mientras aprueba la solicitud de AWS conexión.

8. Selecciona Verificar en AWS.

9. La página Verificar CodeCatalyst el espacio de Amazon se abre en AWS Management Console. Esta es la página de Amazon CodeCatalyst Spaces. Puede que tengas que iniciar sesión para acceder a la página.

En el AWS Management Console, asegúrate de elegir el mismo Región de AWS lugar en el que deseas crear tu espacio.

Para acceder directamente a la página, inicia sesión en Amazon CodeCatalyst Spaces AWS Management Console en <https://console.aws.amazon.com/codecatalyst/home/>.

El token de verificación se introduce automáticamente en el token de verificación. Un aviso de confirmación muestra un mensaje que indica que el token es válido.

10. Seleccione Verificar espacio.

Aparece un mensaje de que la cuenta se ha verificado correctamente para indicar que la cuenta se ha agregado al espacio.

11. Permanece en la página Verify Amazon CodeCatalyst Space. Elija el siguiente enlace: Para añadir funciones de IAM a este espacio, consulte los detalles del espacio.

La página de detalles del CodeCatalyst espacio se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

12. En Funciones de IAM disponibles para CodeCatalyst, selecciona Añadir función de IAM.

Aparece la página Añadir funciones de IAM disponible en la página CodeCatalyst.

13. Seleccione Crear un rol CodeCatalyst de administrador de desarrollo en IAM. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza del rol de desarrollo.

La función de desarrollador es una función de AWS IAM que permite a sus CodeCatalyst flujos de trabajo acceder a AWS recursos como Amazon S3, Lambda y. AWS CloudFormation El rol tendrá un nombre `CodeCatalystWorkflowDevelopmentRole-spaceName` con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#).

14. Elija Crear rol de desarrollo.

15. En la página de conexión, en Funciones de IAM disponibles para CodeCatalyst, consulta la función de desarrollador en la lista de funciones de IAM añadidas a tu cuenta.

16. Selecciona Ir a Amazon CodeCatalyst.
17. En la página de creación de CodeCatalyst, selecciona Crear espacio.

Edición de un espacio

Puedes cambiar la descripción de un espacio para que los usuarios entiendan mejor para qué sirve.

Debe tener el rol de administrador del espacio para editar los detalles del espacio.

La información de esta guía se proporciona para editar los espacios CodeCatalyst que admiten a los usuarios de AWS Builder ID. Para obtener más información sobre los pasos para configurar y administrar un espacio que admita la federación de identidades, consulte [Configuración y administración de CodeCatalyst espacios](#) en la Guía del CodeCatalyst administrador de Amazon.

Para editar la descripción de un espacio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. En la pestaña Configuración del espacio, selecciona Editar. Realice los cambios que desee en la descripción del espacio y, a continuación, seleccione Guardar.

Eliminar un espacio en CodeCatalyst

Puedes eliminar un espacio para eliminar el acceso a todos los recursos del espacio. Debe tener la función de administrador del espacio para eliminar un espacio.

Note

No se puede deshacer la eliminación de un espacio.

Tras eliminar un espacio, todos los miembros del espacio no podrán acceder a los recursos del espacio. También se detendrá la facturación de los recursos de espacio y se detendrán todos los flujos de trabajo que soliciten repositorios de fuentes de terceros.

Note

Los nombres de los espacios deben ser únicos en todos los CodeCatalyst espacios. No puede volver a utilizar los nombres de los espacios eliminados.

La información de esta guía se proporciona para eliminar los espacios CodeCatalyst que admiten a los usuarios de AWS Builder ID. Para obtener más información sobre los pasos para configurar y administrar un espacio que admita la federación de identidades, consulte [Configuración y administración de CodeCatalyst espacios](#) en la Guía del CodeCatalyst administrador de Amazon.

Para eliminar un espacio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Eliminar.
4. Escriba **delete** para confirmar la eliminación.
5. Elija Eliminar.

Note

Si perteneces a más de un espacio, se te redirigirá a la página de descripción general del espacio. Si perteneces a un espacio, se te redirigirá a la página de creación del espacio.

Supervisión de la actividad de un espacio

Para ver los proyectos creados recientemente y las actualizaciones de estado, puedes usar la CodeCatalyst consola para ver un feed de actividades que muestre las actualizaciones de los recursos del espacio.

En el feed de actividades, puedes ver métricas como las ejecuciones fallidas del flujo de trabajo y los proyectos creados.

Para ver la actividad en tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. Seleccione Activity (Actividad).
4. Consulta la información en Actividad.
5. Para filtrar por actividad, selecciona el selector de la esquina superior derecha.
6. Para ver todas las actividades de tu espacio, selecciona Cualquier tipo de actividad.

Cuentas de AWS Administrar un espacio

Puedes usar los recursos de tus CodeCatalyst espacios Cuentas de AWS en Amazon. Para ello, debes configurar una conexión entre el espacio Cuentas de AWS y tu espacio en él CodeCatalyst. Crear una conexión como esta significa que los proyectos y flujos de trabajo de tu CodeCatalyst espacio pueden interactuar con los recursos de tu espacio Cuentas de AWS. Debes crear una conexión para cada una de las que Cuenta de AWS quieras usar con tu CodeCatalyst espacio.

Tras crear una conexión, puede optar por asociarle funciones de AWS IAM.

Temas

- [Añadir un elemento Cuenta de AWS a un espacio](#)
- [Añadir funciones de IAM a las conexiones de cuentas](#)
- [Añadir las funciones de IAM y de conexión de la cuenta a su entorno de implementación](#)

- [Ver las conexiones de la cuenta](#)
- [Eliminar una cuenta de un espacio \(en CodeCatalyst\)](#)

CodeCatalyst Para configurar el uso autorizado, Cuentas de AWS añade las cuentas a su espacio. Al añadirlo Cuentas de AWS a tu CodeCatalyst espacio, puedes dar a los flujos de trabajo de tus proyectos acceso a Cuenta de AWS los recursos y a tu configuración de facturación.

Al agregar una, se Cuenta de AWS crea una conexión que autoriza el CodeCatalyst uso de esta cuenta. Puede utilizar added Cuentas de AWS para hacer lo siguiente:

- Configura la facturación de un CodeCatalyst espacio. Consulta [Gestión de la facturación](#) en la Guía CodeCatalyst del administrador de Amazon.
- Permite CodeCatalyst asumir funciones de IAM para acceder a AWS los recursos e implementarlos Servicios de AWS en la cuenta. Consulte [Cuentas de AWS Administrar un espacio](#).

Las conexiones de la cuenta se crean al completar la autorización con el Cuenta de AWS. Una vez creada la conexión, puede configurar aún más la conexión para que la utilicen los flujos de trabajo y los proyectos añadiendo funciones de IAM.

Añadir un elemento Cuenta de AWS a un espacio

Usas la CodeCatalyst consola y el AWS Management Console para conectar tu espacio a un Cuenta de AWS.

Antes de añadir un Cuenta de AWS elemento a un espacio CodeCatalyst, complete los siguientes requisitos previos:

- Cree Cuenta de AWS y adquiera permisos para crear funciones de AWS IAM en la cuenta a la que desee conectarse.
- Cree el rol o los roles de IAM que desee asociar a la conexión de su cuenta, incluidas las políticas de IAM con los permisos para los roles.
- Adquiera el rol de administrador de Space en el CodeCatalyst espacio en el que desee crear la conexión.

Temas

- [Paso 1: Crear una solicitud de conexión](#)

- [Paso 2: Aceptar una solicitud de conexión de cuenta](#)
- [Paso 3: Revisa una conexión aprobada](#)
- [Paso 4: Añada funciones de IAM a su conexión](#)
- [Próximos pasos: cree funciones de IAM adicionales para la conexión de su cuenta](#)

Paso 1: Crear una solicitud de conexión

Al crear una solicitud de conexión en la CodeCatalyst consola, se genera un token de conexión que puede usar para completar la autorización.

Debe tener el rol de administrador del espacio o usuario avanzado en el CodeCatalyst espacio en el que desee crear la conexión. También debe tener permisos administrativos para el Cuenta de AWS que desee añadir.

Para crear una conexión

1. En el AWS Management Console, asegúrate de haber iniciado sesión con la misma cuenta con la que deseas crear una conexión.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
4. Selecciona Añadir un Cuenta de AWS.
5. En la CodeCatalyst página Associate Cuenta de AWS with Amazon, en Cuenta de AWS ID, introduce el ID de doce dígitos de la cuenta que quieres conectar a tu espacio. Para obtener información sobre cómo encontrar tu Cuenta de AWS ID, consulta [Tu Cuenta de AWS ID y su alias](#).
6. En Amazon CodeCatalyst display name, introduce un nombre de referencia para la cuenta.
7. (Opcional) En la descripción de Connection, introduce una descripción para la cuenta que te ayude a elegir los proyectos a los que se aplicará la cuenta y el rol o los roles.
8. Selecciona Asociar Cuenta de AWS.
9. La página vuelve a la página de Cuenta de AWS detalles, donde aparece un aviso de éxito.

Paso 2: Aceptar una solicitud de conexión de cuenta

Tras enviar una solicitud en la CodeCatalyst consola para conectarse a la suya Cuenta de AWS, deberá trabajar con el AWS administrador para aceptar la solicitud de conexión enviándola con el token de conexión proporcionado.

Asegúrate de tener permisos de administrador en tu cuenta y de haber iniciado sesión AWS Management Console con los mismos permisos Cuenta de AWS para los que estás creando la conexión.

Para aprobar una solicitud de conexión (consola)

1. En AWS Management Console, asegúrate de haber iniciado sesión con la misma cuenta con la que deseas crear una conexión.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
4. En la página de Cuenta de AWS detalles, selecciona Configuración completa en AWS Management Console.
5. La página Verificar CodeCatalyst el espacio de Amazon se abre en AWS Management Console. Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

Para acceder directamente a la página, inicia sesión en Amazon CodeCatalyst Spaces AWS Management Console en <https://console.aws.amazon.com/codecatalyst/home/>.

El token de verificación se introduce automáticamente en el token de verificación. Un mensaje de confirmación muestra un mensaje que indica que el token es válido.

6. (Opcional) En los niveles de pago autorizados, selecciona Autorizar los niveles de pago (Estándar, Enterprise) para activar los niveles de pago en tu cuenta de facturación.

Note

Esto no actualiza el nivel de facturación a un nivel de pago. Sin embargo, esto lo configura Cuenta de AWS para que puedas cambiar el nivel de facturación de tu espacio en cualquier momento. CodeCatalyst Puedes activar los niveles de pago en cualquier momento. Sin realizar este cambio, el espacio solo puede usar el nivel gratuito.

7. Seleccione Verificar espacio.

Aparece un mensaje de confirmación de la cuenta para indicar que la cuenta se ha agregado al espacio.

Paso 3: Revisa una conexión aprobada

Tras obtener la aprobación de una conexión, podrá verla en la consola, junto con las funciones de IAM que le haya agregado.

Para revisar una conexión aprobada

1. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
2. La conexión de la cuenta aparece con la fecha en que se creó.
3. Elige el nombre para mostrar de la cuenta. Aparece la página de Cuenta de AWS detalles.

Paso 4: Añada funciones de IAM a su conexión

Si utiliza un rol de IAM configurado para una acción de CodeCatalyst despliegue, añada el rol a su entorno de despliegue. Para obtener más información, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Próximos pasos: cree funciones de IAM adicionales para la conexión de su cuenta

Tras crear una conexión, puede crear funciones de IAM adicionales para añadirlas. Las funciones de IAM que añada dependen de sus flujos de trabajo. Por ejemplo, una acción de CodeCatalyst creación requiere el rol de CodeCatalyst creación.

Para conectar su cuenta, necesitará el nombre de recurso de Amazon (ARN) de las funciones que creó. Copia el ARN de tu función o funciones tal y como se detalla aquí. Para obtener más información sobre cómo trabajar con los ARN para las funciones de IAM, consulte [Amazon Resource Name \(ARN\)](#).

Para acceder al ARN de su rol de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. En el cuadro de búsqueda, introduzca el nombre del rol que desee añadir.

4. Elija el rol de la lista.

Aparece la página de resumen del rol.

5. En la parte superior, copia el valor del ARN del rol.

Añadir funciones de IAM a las conexiones de cuentas

Parte de la creación de la conexión de tu cuenta incluye añadir el rol o los roles de IAM que quieras usar en los proyectos de tu CodeCatalyst espacio.

Note

Para usar los roles de IAM con una conexión de cuenta, asegúrate de que la política de confianza esté actualizada para usar el principal de CodeCatalyst servicio.

Añada funciones de IAM a una conexión de cuenta (consola)

1. En AWS Management Console, asegúrese de haber iniciado sesión con la misma cuenta que desea administrar.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
4. Elige el nombre CodeCatalyst visible de Amazon de la conexión de tu cuenta y, a continuación, selecciona Administrar roles desde AWS Management Console.

Aparece la página Añadir función de IAM a Amazon CodeCatalyst Space.

5. Realice una de las acciones siguientes:
 - Para crear un rol de servicio que contenga la política de permisos y la política de confianza para el rol de desarrollador, elija Crear un rol de administrador de CodeCatalyst desarrollo en IAM. El rol tendrá un nombre CodeCatalystWorkflowDevelopmentRole-*spaceName* con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#).

Elija Crear rol de desarrollo.

- Para añadir un rol que ya haya creado en IAM, elija Añadir un rol de IAM existente. En Seleccione el rol de IAM existente, elija el rol en la lista desplegable.

Seleccione Add role (Añadir rol).

La página se abre en. AWS Management Console Es posible que deba iniciar sesión para acceder a la página.

6. En el panel de navegación de la página Amazon CodeCatalyst Spaces, selecciona Spaces.

Para acceder directamente a la página, inicia sesión en Amazon CodeCatalyst Spaces AWS Management Console en <https://console.aws.amazon.com/codecatalyst/home/>.

7. Elige la cuenta que has añadido a tu CodeCatalyst espacio. Se muestra la página de conexión.
8. En la página de conexión, en Funciones de IAM disponibles para CodeCatalyst, consulta la lista de funciones de IAM añadidas a tu cuenta. Elija Asociar el rol de IAM a. CodeCatalyst
9. En la ventana emergente Asociar un rol de IAM, en el ARN del rol, introduce el nombre del recurso de Amazon (ARN) del rol de IAM que deseas asociar a tu espacio. CodeCatalyst

En Propósito, elige el propósito de un rol que describa cómo quieres usar el rol en la conexión de tu cuenta. Especifique RUNNER los roles que utilizará para ejecutar acciones en los flujos de trabajo. Especifique SERVICE las funciones que utilizará para acceder a otro servicio.

Puede especificar más de un propósito.

Note

Es obligatorio elegir un propósito para el ARN del rol.

10. Elija Asociar un rol de IAM. Repita estos pasos para funciones de IAM adicionales.

Añadir las funciones de IAM y de conexión de la cuenta a su entorno de implementación

Para acceder a AWS los recursos, como Amazon ECS o AWS Lambda los recursos para las implementaciones, las acciones de CodeCatalyst creación e implementación requieren funciones de IAM con permisos para acceder a esos recursos. Con el rol de administrador de Space o usuario avanzado, puede conectar su CodeCatalyst cuenta al Cuenta de AWS lugar donde se crean sus

recursos. A continuación, añada la función de IAM a la conexión de su cuenta. Para realizar acciones de despliegue, debe añadir la función de IAM a un CodeCatalyst entorno.

Debe añadir las funciones de IAM que desee utilizar con los entornos de implementación de sus proyectos. Al añadir las funciones a la conexión de la cuenta, no se añaden las funciones ni la conexión a los entornos de implementación del proyecto. Para añadir la conexión de la cuenta y las funciones de IAM a su entorno de implementación, asegúrese de que la conexión de la cuenta y las funciones se hayan creado tal y como se detalla en [Paso 4: Añada funciones de IAM a su conexión](#) la sección.

A continuación, utilice la página Entornos de la CodeCatalyst consola para añadir la conexión de su cuenta y su función de IAM a un entorno de despliegue de un proyecto.

Note

Solo se añade una función de IAM a un entorno si la función de IAM se utiliza para una CodeCatalyst acción que requiere una función de IAM. Todas las acciones de flujo de trabajo que requieren funciones de IAM, incluidas las acciones de creación, deben utilizar un entorno. CodeCatalyst

Para añadir las funciones de IAM y conexión de su cuenta a su entorno de implementación

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto con el entorno de despliegue en el que desee añadir la conexión de cuentas y las funciones de IAM.
3. Expanda CI/CD y, a continuación, elija Entornos.
4. Elija su entorno y, a continuación, se mostrarán las pestañas adicionales.
5. Seleccione la pestaña de Cuenta de AWS conexiones. En el nombre de la conexión, se muestran las cuentas que se han agregado al entorno, si las hay.
6. Seleccione Asociar Cuenta de AWS. Aparece la <environment_name>página Asociar Cuenta de AWS con.
7. En Conexión, elija el nombre de la conexión de la cuenta con las funciones de IAM que desee añadir. Seleccione Asociar.

Ver las conexiones de la cuenta

Puedes ver una lista de tus conexiones y ver los detalles de cada conexión.

Debe tener el rol de administrador del espacio o usuario avanzado para administrar las conexiones de su espacio.

Para ver todas las conexiones de un CodeCatalyst espacio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta el espacio con la conexión de la cuenta que deseas ver.
3. Selecciona la pestaña AWS de cuentas.
4. En AWS cuentas, consulta la lista de conexiones de cuentas del espacio, incluidos el ID de cuenta y el estado de cada conexión.

Para ver los detalles de conexión de la cuenta

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
3. En Amazon CodeCatalyst display name, elige el nombre de la conexión. En la página de detalles, consulta la lista de funciones de IAM asociadas a la conexión, junto con otros detalles.


Eliminar una cuenta de un espacio (en CodeCatalyst)

Puedes eliminar una conexión de cuenta que ya no necesites. Para este procedimiento, eliminará una conexión de cuenta que haya agregado anteriormente a su espacio. CodeCatalyst Esto eliminará la conexión de la cuenta de su espacio, siempre que la cuenta no sea la cuenta de facturación del espacio.

Important

Una vez eliminada la conexión de una cuenta, no podrás volver a conectarla. Debe crear una nueva conexión de cuenta y, a continuación, asociar las funciones y los entornos de IAM o configurar la facturación, según sea necesario.

Debe designarse una cuenta de facturación para su CodeCatalyst espacio, incluso si el uso del espacio no superará el nivel gratuito. Antes de poder eliminar un espacio de una cuenta que sea una cuenta de facturación designada, tendrás que añadir otra cuenta para tu espacio. Consulta [Gestión de la facturación](#) en la Guía CodeCatalyst del administrador de Amazon.

 Important

Si bien puedes seguir estos pasos para eliminar una cuenta, no es recomendable. La cuenta también puede configurarse para admitir flujos de trabajo en CodeCatalyst.

Para administrar las conexiones de las cuentas de tu espacio, debes tener el rol de administrador del espacio o de usuario avanzado.

Para eliminar la conexión de una cuenta

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
3. En el nombre para CodeCatalyst mostrar de Amazon, selecciona el selector situado junto a la conexión de cuenta que quieres eliminar.
4. Elija Eliminar Cuenta de AWS. Confirma la eliminación introduciendo el nombre en el campo y, a continuación, selecciona Eliminar.

Aparece un aviso de confirmación y la conexión de la cuenta se elimina de la lista de conexiones.

Gestión de las funciones de IAM para las cuentas conectadas

Los roles se crean en AWS Identity and Access Management (IAM) para la cuenta a la que se quieren añadir. CodeCatalyst Si va a añadir una cuenta de facturación, no necesita crear funciones.

En la suya Cuenta de AWS, debe tener permisos para crear los roles Cuenta de AWS que desee añadir a su espacio. Para obtener más información sobre las funciones y políticas de IAM, incluidas las referencias de IAM y los ejemplos de políticas, consulte. [Identity and Access Management y Amazon CodeCatalyst](#) Para obtener más información sobre la política de confianza y los principios de servicio utilizados en CodeCatalyst, consulte. [CodeCatalyst modelo de confianza](#)

En CodeCatalyst, debe iniciar sesión con el rol de administrador del espacio para completar los pasos necesarios para agregar cuentas (y los roles, si corresponde) a su espacio.

Puedes añadir funciones a las conexiones de tu cuenta mediante uno de los siguientes métodos.

- Para crear un rol de servicio que contenga la política de permisos y la política de confianza del CodeCatalystWorkflowDevelopmentRole-*spaceName*rol, consulte [Rol de CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).
- Para ver un ejemplo de cómo crear un rol y agregar una política para crear un proyecto a partir de un plano, consulte [Crear un rol de IAM y usar la política de CodeCatalyst confianza](#).
- Para ver una lista de ejemplos de políticas de roles que puede usar al crear sus roles de IAM, consulte. [Funciones de IAM para el CodeCatalyst acceso de Amazon a los recursos AWS](#)
- Para ver los pasos detallados para crear roles para las acciones del flujo de trabajo, consulte el siguiente tutorial sobre el flujo de trabajo correspondiente a esa acción:
 - [Tutorial: Carga artefactos a Amazon S3](#)
 - [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#)
 - [Tutorial: Implementación de una aplicación en Amazon ECS](#)
 - [Tutorial: código de Lint mediante una acción GitHub](#)

Temas

- [Rol de CodeCatalystWorkflowDevelopmentRole-*spaceName*](#)
- [Rol de AWSRoleForCodeCatalystSupport](#)
- [Crear un rol de IAM y usar la política de CodeCatalyst confianza](#)

Rol de CodeCatalystWorkflowDevelopmentRole-*spaceName*

El rol de desarrollador se crea como un rol de 1 clic en IAM. Debe tener el rol de administrador de Space o de usuario avanzado en el espacio en el que desee añadir la cuenta. También debe tener permisos administrativos para el Cuenta de AWS que desee añadir.


Antes de iniciar el procedimiento que se indica a continuación, debe iniciar sesión en el AWS Management Console con la misma cuenta que desea añadir a su CodeCatalyst espacio. De lo contrario, la consola devolverá un error de cuenta desconocido.

Para crear y añadir el CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Antes de empezar en la CodeCatalyst consola, abra el espacio y AWS Management Console, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
4. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de AWS detalles.
5. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM a Amazon CodeCatalyst Space se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

6. Elija Crear función CodeCatalyst de administrador de desarrollo en IAM. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza del rol de desarrollo. El rol tendrá un nombreCodeCatalystWorkflowDevelopmentRole-*spaceName*. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#).

 Note

Este rol solo se recomienda para su uso con cuentas de desarrollador y usa la política AdministratorAccess AWS administrada, lo que le da acceso total para crear nuevas políticas y recursos en ella Cuenta de AWS.

7. Selecciona Crear rol de desarrollo.
8. En la página de conexiones, en la sección Funciones de IAM disponibles para CodeCatalyst, consulta la CodeCatalystWorkflowDevelopmentRole-*spaceName* función en la lista de funciones de IAM añadidas a tu cuenta.
9. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Rol de AWSSRoleForCodeCatalystSupport

El rol de soporte se crea como un rol de 1 clic en IAM. Debe tener el rol de administrador del espacio o usuario avanzado en el espacio en el que desee añadir la cuenta. También debe tener permisos administrativos para el Cuenta de AWS que desee añadir.

Antes de iniciar el procedimiento que se indica a continuación, debe iniciar sesión en el AWS Management Console con la misma cuenta que desea añadir a su CodeCatalyst espacio. De lo contrario, la consola devolverá un error de cuenta desconocido.

Para crear y añadir el CodeCatalyst AWSSRoleForCodeCatalystSupport

1. Antes de empezar en la CodeCatalyst consola, abra el espacio y AWS Management Console, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:
3. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de AWS detalles.
4. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM a Amazon CodeCatalyst Space se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Puede que tengas que iniciar sesión para acceder a la página.

5. En los detalles del CodeCatalyst espacio, selecciona Add CodeCatalyst Support role. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza para el rol de desarrollo preliminar. El rol tendrá un nombre AWSSRoleForCodeCatalystSupport con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Entender la función AWSSRoleForCodeCatalystSupport de servicio](#).
6. En la página Añadir función para CodeCatalyst Support, deje seleccionada la opción predeterminada y, a continuación, elija Crear función.
7. En Funciones de IAM disponibles para CodeCatalyst, consulta la CodeCatalystWorkflowDevelopmentRole-*spaceName* función en la lista de funciones de IAM añadidas a tu cuenta.
8. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Crear un rol de IAM y usar la política de CodeCatalyst confianza

Las funciones de IAM que se van a utilizar CodeCatalyst con Cuenta de AWS las conexiones deben configurarse para utilizar la política de confianza que se proporciona aquí. Siga estos pasos para crear un rol de IAM y adjuntar una política que le permita crear proyectos a partir de planos integrados. CodeCatalyst

Como alternativa, puede crear un rol de servicio que contenga la política de permisos y la política de confianza del CodeCatalystWorkflowDevelopmentRole-*spaceName* rol. Para obtener más información, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles y, a continuación, Crear rol.
3. Elige Política de confianza personalizada.
4. En el formulario de política de confianza personalizada, pegue la siguiente política de confianza.

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/
**
          }
        }
      }
    ]
```

5. Elija Siguiente.

6. En Añadir permisos, busque y seleccione una política personalizada que ya haya creado en IAM.
7. Elija Siguiente.
8. En Nombre del rol, introduzca un nombre para el rol, por ejemplo: `codecatalyst-project-role`
9. Seleccione Crear rol.
10. Copie el nombre de recurso de Amazon (ARN) del rol. Deberá proporcionar esta información cuando añada el rol a la conexión o el entorno de su cuenta.

Gestión de los usuarios del espacio

Para administrar los miembros de un espacio, ve, agrega, elimina o cambia los roles de los usuarios que se unen al espacio.

La información de esta guía se proporciona para invitar y administrar usuarios en espacios CodeCatalyst que admiten usuarios de AWS Builder ID. Para obtener más información sobre los pasos para configurar y administrar un espacio que admita la federación de identidades, consulte [Configuración y administración de CodeCatalyst espacios](#) en la Guía del CodeCatalyst administrador de Amazon.

Ver los miembros de un espacio

Puede ver los usuarios de su espacio, incluida la información sobre sus nombres para mostrar, sus alias y la función que desempeñan en el espacio. Los miembros de un espacio tienen tres funciones:

- **Administrador del espacio:** este rol tiene todos los permisos CodeCatalyst, incluida la creación de proyectos. Asigne este rol únicamente a los usuarios que necesiten administrar todos los aspectos de un espacio, por ejemplo, acceder a todos los proyectos del espacio.

No puede cambiar este rol más adelante sin eliminar primero al usuario. Para obtener más información, consulte [Función de administrador de espacios](#).

- **Usuario avanzado:** este rol es el segundo más poderoso en los CodeCatalyst espacios de Amazon, pero no tiene acceso a los proyectos de un espacio. Está diseñado para usuarios que necesitan poder crear proyectos en un espacio y ayudar a administrar los usuarios y los recursos del espacio. Para obtener más información, consulte [Rol de usuario avanzado](#).
- **Acceso limitado:** esta función se asigna de forma predeterminada a los usuarios que se unen al espacio aceptando invitaciones a los proyectos del espacio. A los miembros del proyecto se

les asigna un rol en un proyecto. Para obtener información sobre la gestión de los miembros del proyecto, consulte [Gestión de los miembros del proyecto](#).

En la tabla de administradores de espacios se muestran los usuarios con la función de administrador de espacios. Estos usuarios no aparecen en los miembros del espacio porque se les asigna automáticamente (implícitamente) a todos los proyectos del espacio y no tienen ningún rol en ningún proyecto.

La tabla de miembros del espacio muestra todos los miembros del espacio que tienen un rol en un proyecto y no tienen el rol de administrador del espacio.

Los usuarios se muestran en función de si el usuario tiene la función de administrador del espacio de la CodeCatalyst siguiente manera:

- Un usuario con el rol de administrador del espacio que luego acepte la invitación y el rol de un proyecto no aparecerá en la tabla de miembros del espacio, situada debajo de los espacios, ni en la tabla de miembros del proyecto, en la sección de proyectos. Seguirán apareciendo en la tabla de administradores del espacio en ambos lugares. En cada proyecto, todos los usuarios con el rol de administrador del espacio se muestran en la tabla de administradores del espacio del proyecto correspondiente a ese proyecto.
- El usuario que acepta una invitación al proyecto para unirse con un rol de proyecto se agrega al espacio con el rol de acceso limitado. Si el rol del usuario cambia posteriormente al de administrador del espacio, también pasará de la tabla de miembros del espacio a la tabla de administradores del espacio. En el proyecto, el usuario pasará de la tabla de miembros del proyecto a la tabla de administradores del espacio.

Para ver los usuarios y los roles de tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

 Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Miembros.

Los usuarios que son miembros del espacio se muestran en la tabla de miembros del espacio.

i Tip

Si tiene el rol de administrador del espacio, puede ver los proyectos a los que ha sido invitado directamente. Ve a la configuración del proyecto y, a continuación, selecciona Mis proyectos.

En la columna Estado, los siguientes son valores válidos:

- CodeCatalyst Invitado: se envió la invitación, pero el usuario aún no la ha aceptado o rechazado.
- Miembro: el usuario ha aceptado la invitación.

Invitar a un usuario directamente a un espacio

Puedes invitar a los usuarios directamente a tu CodeCatalyst espacio. Esto resulta útil cuando quieres invitar a ese usuario para que te ayude a administrar el espacio asignándole el rol de administrador del espacio o usuario avanzado. Asignar uno de esos roles a otros usuarios puede ayudarte a distribuir las responsabilidades de administrar el espacio entre más personas sin tener que invitarlos a ningún proyecto.

i Note

Debes tener el rol de administrador del espacio o de usuario avanzado para poder invitar a miembros.

En la tabla de administradores del espacio se muestran los usuarios con el rol de administrador del espacio. Estos usuarios no aparecen en la tabla de miembros del espacio porque se les asigna automáticamente (implícitamente) a todos los proyectos del espacio y no tienen ningún rol en ningún proyecto.

Los miembros que aceptan una invitación a un proyecto se añaden al espacio de forma predeterminada. La tabla de miembros del proyecto muestra todos los miembros del espacio que tienen un rol en un proyecto.

Para obtener más información sobre cómo aceptar una invitación e iniciar sesión por primera vez, consulte [Configuración CodeCatalyst](#).

Para invitar a un usuario a tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.
3. Selecciona Configuración y, a continuación, selecciona Miembros.
4. Elija Invitar.
5. Introduce el correo electrónico de la persona a la que quieres invitar a unirse a tu espacio. En Función, elige la función que quieres asignar a ese usuario en el espacio.
6. Elige Invitar

Cancelar una invitación a un espacio

Si quieres cancelar una invitación para unirse a un espacio que enviaste recientemente y aún no se ha aceptado, puedes cancelarla.

Para gestionar las invitaciones a un espacio, debes tener el rol de administrador del espacio o de usuario avanzado.

Para cancelar la invitación de un miembro del espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Miembros.
4. Compruebe que el miembro tiene el estado Invitado.

Note

Solo puedes cancelar una invitación que aún no se haya aceptado.

5. Elige la opción situada junto a la fila con el miembro invitado y, a continuación, selecciona Cancelar la invitación.
6. Aparece una ventana de confirmación. Selecciona Cancelar la invitación para confirmarla.

Cambiar el rol de un miembro del espacio

Puede cambiar el rol asignado a un miembro de su espacio. Debe tener el rol de administrador del espacio para cambiar el rol de un usuario en el espacio.

En la tabla de administradores del espacio se muestran los usuarios con el rol de administrador del espacio. Estos usuarios no aparecen en la tabla de miembros del espacio porque se asignan automáticamente (implícitamente) a todos los proyectos del espacio.

Para cambiar el rol de un usuario en tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Miembros.
4. En la tabla de miembros de Space, elige el usuario cuyo rol quieres cambiar. Selecciona Cambiar rol.

Eliminar un elemento del espacio

Puedes eliminar a un miembro de tu espacio cuando no necesite acceder a ninguno de los recursos del espacio. Debe tener la función de administrador del espacio para eliminar a un miembro de un espacio.

En la tabla de administradores del espacio se muestran los usuarios con el rol de administrador del espacio. Estos usuarios no aparecen en la tabla de miembros del espacio porque se les asigna automáticamente (implícitamente) a todos los proyectos del espacio y no tienen ningún rol en ningún proyecto. Solo puedes eliminar directamente a un miembro de tu espacio en esta tabla.

Para eliminar un usuario de la tabla de miembros del proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio.

Tip

Si pertenece a más de un espacio, elija un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Miembros.
4. Elija el usuario en la tabla de miembros del proyecto. Elija Eliminar.

Note

Al eliminar a un miembro del espacio, se eliminará al usuario de todos los proyectos del espacio, junto con los permisos asociados a los recursos de esos proyectos.

Eliminar o cambiar el rol de un usuario con el rol de administrador de Space

Puede eliminar o cambiar el rol de un usuario con el rol de administrador del espacio.

Debe tener la función de administrador del espacio para eliminar de un espacio a un usuario con la función de administrador del espacio. Al cambiar la función de un usuario por la de administrador de espacios, básicamente se elimina al usuario de la tabla de administradores de espacios. Si ese usuario no tiene una función de proyecto en ningún proyecto del espacio, si se le quita la función de administrador del espacio, se eliminará al usuario del espacio.

Note

Como usuario con el rol de administrador de espacios, no puedes eliminarte a ti mismo. Póngase en contacto con otro usuario con el rol de administrador de Space.

Para eliminar un usuario con el rol de administrador del espacio de la tabla de miembros del espacio

Note

Los usuarios que no se hayan agregado explícitamente a un proyecto no tienen ningún rol en el proyecto (administrador del proyecto o colaborador). Si la función de administrador del espacio es la única función del usuario, el usuario se eliminará por completo del espacio.

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio en el que desee eliminar o cambiar el rol de un usuario con el rol de administrador del espacio.
3. Selecciona Configuración y, a continuación, selecciona Miembros.
4. Consulta el estado de la invitación de la lista de miembros y asegúrate de que la lista no contenga ninguna invitación pendiente no autorizada al espacio (el estado es Invitado).

Important

Antes de eliminar a un usuario con el rol de administrador del espacio, debes comprobar que no se ha iniciado ninguna invitación pendiente.

5. Seleccione la pestaña Miembros. En la tabla de administradores del espacio, elija el usuario y, a continuación, elija Eliminar.

En el cuadro de diálogo Eliminar miembro, realice una de las siguientes acciones.

- Seleccione la opción para eliminar únicamente la función de administrador de Space del usuario. Elija Eliminar.

Important

Si el usuario no tiene ningún otro rol asignado, al cambiar el rol de administrador del espacio, se elimina al usuario del espacio.

- Elija la opción de eliminar un usuario con el rol de administrador del espacio y de todos sus proyectos. Elija Eliminar.
6. Actualice la pestaña Miembros. El usuario se añade automáticamente a la lista de miembros del proyecto en cualquier proyecto al que haya pertenecido a través de funciones de proyecto. Si el

rol de administrador del espacio era el único rol del usuario, el usuario se eliminará por completo del espacio.

Gestión de equipos

Tras crear un espacio, puedes añadir equipos. Los equipos te permiten agrupar a los usuarios para que puedan compartir los permisos y gestionar los proyectos, el seguimiento de los problemas, las funciones y los recursos CodeCatalyst.

Debes tener el rol de administrador de Space para administrar los equipos.

Temas

- [Crear un equipo](#)
- [Ver un equipo](#)
- [Administrar el rol de espacio de un equipo](#)
- [Administrar un rol de proyecto para un equipo](#)
- [Añadir un usuario a un equipo directamente](#)
- [Eliminar directamente a un usuario de un equipo](#)
- [Añadir un grupo de SSO a un equipo](#)
- [Eliminar un equipo](#)

Crear un equipo

Un equipo puede tener permisos de rol, como el de usuario avanzado, en un espacio. Un equipo también puede tener permisos de proyecto, como el de administrador de proyectos, en un proyecto. Los equipos se pueden asociar a muchos proyectos con funciones diferentes para cada proyecto. Puedes gestionar equipos en los que los miembros del equipo sean usuarios individuales para un espacio de AWS Builder ID o grupos de SSO para un espacio que admita la federación de identidades.

En la página de miembros para los usuarios del espacio y del proyecto, los usuarios pueden tener varios roles. Los usuarios con varios roles mostrarán un indicador cuando tengan varios roles y aparecerán primero con el rol con más permisos.

Note

Si su espacio admite la federación de identidades, debe tener ya configurados sus usuarios de SSO o sus grupos de SSO en el Centro de identidades de IAM.

La forma en que gestiones a los miembros del equipo depende de cómo añadas y elimines usuarios. Existen dos opciones para administrar a los miembros del equipo:

- **Añadir usuarios directamente:** puede añadir o eliminar usuarios de forma individual. Por ejemplo, para añadir usuarios a un equipo, elija usuarios de AWS Builder ID o usuarios de SSO que ya estén configurados en el Centro de identidades de IAM. Si decide administrar los miembros del equipo añadiendo usuarios de AWS Builder ID o usuarios de SSO directamente, la opción de usar grupos de SSO ya no estará disponible.
- **Utilice grupos de SSO:** los miembros del equipo se administran mediante grupos de SSO que ya están configurados en el Centro de identidades de IAM. Si decide administrar los miembros del equipo mediante grupos de SSO, la opción de añadir usuarios directamente ya no estará disponible.

Debes tener el rol de administrador de Space para administrar los equipos.

Para crear un equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Selecciona Configuración y, a continuación, selecciona Equipos.
3. Selecciona Crear equipo.
4. En Nombre del equipo, introduce un nombre descriptivo para tu equipo.

Note


El nombre del equipo debe ser único en tu espacio.

(Opcional) En la descripción del equipo, introduce una descripción para tu equipo.

5. En Función de espacio, elige una función de la lista de funciones de espacio disponibles CodeCatalyst que quieras asignar al equipo. Todos los miembros del equipo heredarán el rol.


- Administrador del espacio: para obtener más información, consulte [Función de administrador de espacios](#).
 - Acceso limitado: para obtener más información, consulte [Función de acceso limitado](#).
 - Usuario avanzado: para obtener más información, consulte [Rol de usuario avanzado](#).
6. En Pertenencia al equipo, elige una de las siguientes opciones para elegir el método de añadir miembros al equipo.
- Selecciona Añadir miembros directamente para gestionar los usuarios de forma individual. Esto incluye agregar usuarios de AWS Builder ID para un espacio o agregar usuarios de SSO para un espacio que admita la federación de identidades.
 - Elija Utilizar grupos de SSO para elegir los grupos de SSO que ya haya configurado en el Centro de identidades de IAM.

En Grupos de SSO, seleccione la casilla situada junto a los grupos que desee añadir. Puedes añadir hasta cinco grupos de SSO.

 Note

No puedes cambiar esto más adelante. Si decides administrar los miembros del equipo añadiendo usuarios de AWS Builder ID o usuarios de SSO directamente, la opción de usar grupos de SSO dejará de estar disponible. Si decides administrar los miembros del equipo mediante grupos de SSO, la opción de añadir usuarios directamente ya no estará disponible.

7. Seleccione Crear.

 Note

Si decides usar grupos de SSO, ten en cuenta que los usuarios del grupo de SSO no se eliminan al crear el equipo. Los usuarios deberán haber iniciado sesión CodeCatalyst antes de que aparezcan en la lista.

Ver un equipo

En CodeCatalyst, puedes ver los proyectos y las funciones de tu equipo. En la página de miembros, puedes ver las funciones del proyecto y una lista de usuarios. En el caso de los equipos de tipo grupo SSO, también podrás ver una lista de los grupos de SSO asociados al equipo.

Para ver un equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Selecciona Configuración y, a continuación, selecciona Equipos.
3. En el rol de espacio, consulta el rol asignado al equipo para este espacio.
4. En la pestaña Funciones del proyecto, consulta el proyecto y la función de proyecto asignada al equipo para cada CodeCatalyst proyecto en el espacio en el que se ha añadido al equipo como miembro (solo en el caso de un espacio con ID de AWS constructor).
5. En la pestaña Miembros, consulta la lista de miembros asignados al equipo.
6. En la pestaña Grupos de SSO, consulta la lista de grupos de SSO asignados al equipo (para un espacio que solo admite la federación de identidades).

Administrar el rol de espacio de un equipo

Un equipo puede tener permisos de rol, como el de usuario avanzado, en un espacio. Puedes cambiar el rol de un equipo en el espacio, pero ten en cuenta que todos los miembros del equipo heredarán esos permisos.

Debes tener el rol de administrador del espacio para gestionar los equipos.

Cambiar el rol de un equipo en el espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Selecciona Configuración y, a continuación, selecciona Equipos.
3. En Acciones, selecciona Cambiar rol en el espacio. Puede cambiar el rol del espacio por uno de los siguientes. Esto cambia el rol de todos los miembros del equipo.
 - Administrador del espacio: para obtener más información, consulte [Función de administrador de espacios](#).
 - Acceso limitado: para obtener más información, consulte [Función de acceso limitado](#).

- Usuario avanzado: para obtener más información, consulte [Rol de usuario avanzado](#).

4. Seleccione Guardar.

Administrar un rol de proyecto para un equipo

Un equipo CodeCatalyst es similar a un usuario en el sentido de que los miembros del equipo pueden tener permisos de rol, como los de administrador del proyecto, en un proyecto. Se aplicará un cambio de rol al equipo y todos los miembros del equipo heredarán esos permisos. Puedes elegir un rol para cada proyecto que se otorgará automáticamente al equipo.

Debes tener el rol de administrador del espacio para gestionar los equipos.

Para añadir o cambiar un rol de proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Seleccione Configuración y, a continuación, seleccione Equipos.
3. Seleccione la pestaña Funciones del proyecto.
4. Para cambiar un rol, elija el selector situado junto al proyecto en esta lista y, a continuación, elija Cambiar rol. Para agregar un rol, elija Agregar rol de proyecto. En Proyecto, elija el proyecto que desee añadir y, en Función, elija la función. Elige uno de los roles de proyecto disponibles:
 - Administrador del proyecto: para obtener más información, consulte [Función de administrador del proyecto](#).
 - Colaborador: para obtener más información, consulte [Función de colaborador](#).
 - Revisor: para obtener más información, consulte [Función de revisor](#).
 - Solo lectura: para obtener más información, consulte [Función de solo lectura](#).
5. Seleccione Guardar.

Para eliminar un rol de proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Seleccione Configuración y, a continuación, seleccione Equipos.
3. Seleccione la pestaña Funciones del proyecto.
4. Elige el rol que deseas eliminar.

⚠ Important

Al eliminar un rol de un equipo, se eliminan los permisos asociados para todos los usuarios del equipo.

5. Seleccione Guardar.

Añadir un usuario a un equipo directamente

Puedes añadir miembros del equipo a tu equipo. Al añadir un usuario, el nuevo usuario heredará los permisos de todos los roles existentes en el equipo.

Tanto si tu espacio está configurado para el soporte de usuarios de AWS Builder ID como para la federación de identidades, puedes configurarlo para añadir usuarios directamente.

i Note

Si tu espacio está configurado para administrar a los miembros del equipo mediante grupos de SSO, la opción de usar Añadir usuarios directamente no está disponible. Para usar grupos de SSO, consulta [Añadir un grupo de SSO a un equipo](#)

Debe tener el rol de administrador del espacio para administrar los equipos.

Para añadir un usuario directamente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Seleccione Configuración y, a continuación, seleccione Equipos.
3. Seleccione la pestaña Miembros.
4. Seleccione Añadir miembro.

i Note

Los usuarios que se añadan a un equipo deben ser ya miembros de un espacio. No puedes añadir ni invitar a un miembro del equipo que no sea miembro del espacio.

5. Elige un usuario en el campo desplegable y, a continuación, selecciona Guardar. Elija los usuarios de AWS Builder ID o los usuarios de SSO que ya estén configurados en el Centro de identidades de IAM.

Eliminar directamente a un usuario de un equipo

Puedes eliminar miembros del equipo de tu equipo. El usuario ya no heredará todos los permisos. Puedes volver a añadir al usuario al equipo más adelante.

Note

Al eliminar a un miembro del equipo, se eliminarán los permisos asociados para el usuario de todos los proyectos y recursos del espacio.

Debes tener el rol de administrador del espacio para administrar los equipos.

Para eliminar un miembro del equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Selecciona Configuración y, a continuación, selecciona Equipos.
3. Seleccione la pestaña Miembros.
4. Selecciona el selector situado junto al usuario que quieres eliminar y, a continuación, selecciona Eliminar.
5. Escribe eliminar en el campo de entrada y, a continuación, selecciona Eliminar.

Añadir un grupo de SSO a un equipo

Si tu espacio está configurado como un espacio con usuarios y grupos de SSO gestionados en el Centro de Identidad de IAM, puedes añadir un grupo de SSO que se unirá al espacio como un equipo independiente.

Note

Si decide administrar los miembros del equipo añadiendo usuarios de AWS Builder ID o usuarios de SSO directamente, la opción de usar grupos de SSO no está disponible. Para añadir usuarios directamente, consulte. [Añadir un usuario a un equipo directamente](#)

Debe tener el rol de administrador de Space para administrar los equipos.

Para añadir un grupo de SSO como equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la página de tu espacio, selecciona Teams. Selecciona la pestaña Grupos de SSO.
3. Elige los grupos de SSO que quieres añadir. Puedes añadir hasta cinco grupos de SSO.

Eliminar un equipo

Puedes eliminar un equipo que ya no necesites.

Note

Al eliminar un equipo, se eliminarán los permisos asociados para todos los miembros del equipo de todos los proyectos y recursos del espacio.

Debes tener el rol de administrador del espacio para administrar los equipos.

Eliminar un equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Selecciona Configuración y, a continuación, selecciona Equipos.
3. En Acciones, selecciona Eliminar equipo. Esto cambia la función de todo el equipo.
4. Elija Eliminar.

Gestión de los recursos de la máquina

Un recurso de máquina representa tu identidad con respecto al recurso autorizado cuando accedes a CodeCatalyst través del SSO. Los recursos de la máquina se utilizan para conceder permisos a los recursos del espacio, como los planos y los flujos de trabajo. Puede ver los recursos de máquina de su espacio y puede optar por habilitar o deshabilitar los recursos de máquina para su espacio. Por ejemplo, es posible que desee deshabilitar un recurso de máquina para administrar el acceso y volver a habilitarlo más adelante.

Estas operaciones están disponibles para los recursos de máquina en los casos en que sea necesario revocar o deshabilitar un recurso de máquina. Por ejemplo, si sospecha que las credenciales podrían estar comprometidas, puede deshabilitar el recurso de la máquina. Por lo general, no será necesario utilizar estas operaciones.

Debe tener el rol de administrador del espacio para ver esta página y administrar los recursos de la máquina a nivel de espacio.

Temas

- [Ver los recursos de la máquina](#)
- [Desactive los recursos de la máquina](#)
- [Habilite los recursos de la máquina](#)

Ver los recursos de la máquina

Puede ver una lista de los recursos de la máquina que se utilizan en su espacio.

Debe tener el rol de administrador del espacio para administrar los recursos de la máquina.

Para ver los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Ve a tu espacio y, a continuación, selecciona Configuración. Elige Recursos de la máquina.
3. En el menú desplegable, elija Acción de flujo de trabajo para ver solo los recursos de máquina para los flujos de trabajo. Elija Blueprint para ver solo los recursos de máquina para los blueprints.

También puede filtrar por nombre mediante el campo Filtro.

Desactive los recursos de la máquina

Puede optar por deshabilitar los recursos de la máquina que se utilizan en su espacio.

Important

Al deshabilitar los recursos de la máquina, se eliminarán todos los permisos de todos los blueprints o flujos de trabajo asociados en el espacio.

Debe tener el rol de administrador del espacio para administrar los recursos de la máquina.

Para deshabilitar los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Ve a tu espacio y, a continuación, selecciona Configuración. Elige Recursos de la máquina.
3. Elija una de las siguientes opciones.

Important

Al deshabilitar los recursos de la máquina, se eliminarán todos los permisos de todos los blueprints o flujos de trabajo asociados en el espacio.

- Para deshabilitar uno por uno, elija el selector situado junto a uno o más recursos de máquina que desee deshabilitar. Elija Desactivar y, a continuación, elija Este recurso.
- Para deshabilitar todos los recursos, selecciona Desactivar y, a continuación, selecciona Todos los recursos.
- Para deshabilitar todas las acciones del flujo de trabajo, selecciona Desactivar y, a continuación, selecciona Todas las acciones del flujo de trabajo.
- Para deshabilitar todos los esquemas, seleccione Desactivar y, a continuación, elija Todos los esquemas.

Habilite los recursos de la máquina

Puede optar por habilitar los recursos de máquina que estén en uso en su espacio y que estén deshabilitados.

Debe tener el rol de administrador del espacio para administrar los recursos de la máquina.

Para habilitar los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Ve a tu espacio y, a continuación, selecciona Configuración. Elige Recursos de la máquina.
3. Elija una de las siguientes opciones.
 - Para activarlos de forma individual, seleccione el selector situado junto a uno o más recursos de máquina que desee activar. Elija Habilitar y, a continuación, elija Este recurso.
 - Para habilitar todos los recursos, elija Habilitar y, a continuación, elija Todos los recursos.
 - Para habilitar todas las acciones del flujo de trabajo, elija Habilitar y, a continuación, elija Todas las acciones del flujo de trabajo.
 - Para habilitar todos los esquemas, seleccione Habilitar y, a continuación, elija Todos los esquemas.

Administración de entornos de desarrollo para un espacio

Todos los entornos de desarrollo se crean como parte de un proyecto dentro de un espacio. Los miembros de Space pueden crear sus propios entornos de desarrollo dentro de un proyecto a nivel del repositorio de origen. Luego, los administradores de espacios pueden usar la CodeCatalyst consola de Amazon para ver, editar, eliminar y detener entornos de desarrollo en nombre de los miembros del espacio. En resumen, los administradores de espacios mantienen los entornos de desarrollo a nivel de espacio.

Consideraciones para administrar entornos de desarrollo

- Debe tener el rol de administrador de espacios para ver la página Entornos de desarrollo en Configuración y para administrar los entornos de desarrollo a nivel de espacio.
- Los miembros de Space administran los entornos de desarrollo que crean en los proyectos a través de sus CodeCatalyst cuentas. Cuando administras los entornos de desarrollo como administrador del espacio, mantienes estos recursos en nombre de los miembros del espacio.
- Los entornos de desarrollo tienen de forma predeterminada una configuración de procesamiento y almacenamiento específica. Para obtener información sobre la facturación y las tarifas para actualizar tu configuración, consulta la [página de CodeCatalyst precios de Amazon](#).

Para ver otras consideraciones sobre los entornos de desarrollo, como detener las instancias en ejecución, configurar la configuración informática predeterminada, actualizar la informática, incurrir en costes y configurar los tiempos de espera, consulte [Entornos de desarrollo en CodeCatalyst](#)

Temas

- [Visualización de los entornos de desarrollo de su espacio](#)
- [Edición de un entorno de desarrollo para tu espacio](#)
- [Detener un entorno de desarrollo para su espacio](#)
- [Eliminar un entorno de desarrollo para tu espacio](#)

Visualización de los entornos de desarrollo de su espacio

Puedes ver el tipo, el estado y los detalles de todos los entornos de desarrollo de tu espacio. Para obtener más información sobre la creación y ejecución de entornos de desarrollo, consulte [Creación de un entorno de desarrollo](#).

Debe tener el rol de administrador de espacios para ver esta página y administrar los entornos de desarrollo a nivel de espacio.

Para ver los entornos de desarrollo de tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Entornos de desarrollo.

La página muestra todos los entornos de desarrollo de tu espacio. Puede ver el nombre del recurso, el alias del recurso, si corresponde, el tipo de IDE, el cómputo y el almacenamiento predeterminados o configurados y el tiempo de espera configurado para cada entorno de desarrollo.

Edición de un entorno de desarrollo para tu espacio

Puede editar la configuración de un entorno de desarrollo, como el tiempo de espera configurado, si lo hubiera, para que un entorno de desarrollo inactivo deje de funcionar. Para obtener más información sobre la edición de un entorno de desarrollo, consulte [Edición de un entorno de desarrollo](#).

Debe tener el rol de administrador de espacios para ver esta página y administrar los entornos de desarrollo a nivel de espacio.

Para editar los entornos de desarrollo de tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Entornos de desarrollo.
4. Selecciona el selector situado junto al entorno de desarrollo que quieres gestionar. Elija Editar.
5. Realice los cambios que desee en el tiempo de espera de procesamiento o inactividad del entorno de desarrollo.
6. Elija Guardar.

Detener un entorno de desarrollo para su espacio

Puede detener un entorno de desarrollo en ejecución antes de que quede inactivo si el entorno de desarrollo está configurado para tener un tiempo de espera. De lo contrario, un entorno de desarrollo con un tiempo de espera transcurrido ya estará detenido. Para obtener más información sobre cómo detener un entorno de desarrollo, consulte [Detención de un entorno de desarrollo](#)

Debe tener el rol de administrador de espacios para ver esta página y administrar los entornos de desarrollo a nivel de espacio.

Para detener un entorno de desarrollo en tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. Navega hasta tu CodeCatalyst espacio.

 Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Entornos de desarrollo.
4. Selecciona el selector situado junto al entorno de desarrollo que quieres gestionar. Elija Detener.

Eliminar un entorno de desarrollo para tu espacio

Puedes eliminar un entorno de desarrollo que ya no sea necesario o que ya no tenga propietario. Para obtener más información sobre las consideraciones a la hora de eliminar un entorno de desarrollo, consulte [Eliminación de un entorno de desarrollo](#).

Debe tener la función de administrador de espacios para ver esta página y gestionar los entornos de desarrollo a nivel de espacio.

Para eliminar los entornos de desarrollo de tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

 Tip

Si perteneces a más de un espacio, elige uno en la barra de navegación superior.

3. Selecciona Configuración y, a continuación, selecciona Entornos de desarrollo.
4. Selecciona el selector situado junto al entorno de desarrollo que quieres gestionar. Elija Eliminar. Para confirmar, escriba `ydelete`, a continuación, seleccione Eliminar.

Cuotas de espacios en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de los espacios en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Número de espacios activos por usuario y por Región de AWS	Cinco
Número de creaciones de espacios por región, mes y usuario	Cinco
Descripciones de espacios	<p>Las descripciones de los espacios son opcionales. Si se especifica, deben tener una longitud de entre 0 y 200 caracteres. Pueden contener cualquier combinación de letras, números, espacios, puntos, guiones bajos, comas, guiones y los siguientes caracteres especiales:</p> <p>? & \$ % + = / \ ; : \n \t \r</p>
Nombres de espacios	<p>Los nombres de los espacios deben ser únicos en todos ellos CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados. Los nombres de los espacios deben tener una longitud de entre 3 y 63 caracteres. También deben empezar por un carácter alfanumérico. Los nombres de los espacios pueden contener cualquier combinación de letras, números, puntos, guiones bajos y guiones. No pueden contener ninguno de los siguientes caracteres:</p> <p>! ? @ # \$ % ^ & * () + = { } [] \ > < ~ ` ' " ; :</p>

Proyectos en CodeCatalyst

Utiliza los proyectos en Amazon CodeCatalyst para establecer un espacio de colaboración en el que los equipos de desarrollo puedan realizar tareas de desarrollo con repositorios y flujos de trabajo de integración continua y entrega continua (CI/CD) compartidos. Al crear un proyecto, puede añadir, actualizar o eliminar recursos. También puedes supervisar el progreso del trabajo de tu equipo. Puede tener varios proyectos dentro de un espacio.

Los espacios CodeCatalyst están compuestos por proyectos. Puedes ver todos los proyectos de tu espacio, pero solo puedes usar los proyectos de los que eres miembro. Cuando creas un proyecto, se generan las funciones predeterminadas para el proyecto, que asignas a los usuarios que invitas a tu proyecto.

- Cualquier persona asignada al proyecto con un rol de proyecto, como el rol de colaborador, puede acceder a los recursos del proyecto, como un repositorio de fuentes.
- Cualquier persona que ocupe el puesto de administrador del espacio o administrador del proyecto puede enviar invitaciones para unirse a un proyecto.
- Los usuarios con el rol de administrador del proyecto pueden realizar un seguimiento de la actividad, el estado y otros ajustes en los recursos compartidos.
- Los usuarios con el rol de acceso limitado pueden gestionar la asignación de funciones, correcciones de código y pruebas a los proyectos como parte de los flujos de trabajo de CI/CD.

Los flujos de trabajo se utilizan para crear, probar y lanzar o actualizar aplicaciones como una canalización de CI/CD. Puede ensamblar flujos de trabajo añadiendo acciones que transfieran los artefactos de origen y funcionen con ellos. Cuando ejecutas acciones, los recursos de la nube de tu proyecto se utilizan para proporcionar capacidad de procesamiento bajo demanda para las acciones de tu flujo de trabajo. Puede configurar más flujos de trabajo de CI/CD en función de la actividad y el resultado que desee configurar. Por ejemplo, puede crear un flujo de trabajo solo para las acciones de creación y prueba, en el que pueda ver los resultados de las pruebas y completar el flujo de trabajo sin necesidad de una implementación mientras corrige los errores. A continuación, puede crear otro flujo de trabajo para crear e implementar la aplicación en un entorno provisional.

Al crear un proyecto, puede usar un plano para crear un proyecto que contenga código de ejemplo y cree recursos, o puede empezar con un proyecto vacío. Si crea un proyecto mediante un esquema, el esquema que elija determinará los recursos que se añadirán al proyecto y las herramientas que

se CodeCatalyst crearán o configurarán para que pueda realizar un seguimiento de los recursos del proyecto y utilizarlos. Puede añadir o eliminar recursos manualmente después de haber creado un proyecto. Los siguientes recursos pueden crearlos o configurarlos CodeCatalyst:

- **Problemas:** realiza un seguimiento del trabajo relacionado con su proyecto en un registro distinto denominado problemas. Puedes crear problemas con las funciones, las tareas, los errores y cualquier otro elemento relacionado con tus proyectos.
- **Notificaciones:** para configurar las notificaciones, elige el recurso que quieres ver, el evento que quieres ver y el cliente o correo electrónico de destino en el que quieres recibir las notificaciones.
- **Búsqueda:** puedes buscar código, problemas, usuarios, solicitudes de cambios y paquetes en tu proyecto. Puedes buscar en un solo proyecto o en todos los proyectos.
- **Repositorio de código fuente:** trabajas con tu código fuente en un repositorio para tu proyecto. Cuando realizas un cambio en el código fuente o fusionas una solicitud de incorporación de cambios en una rama específica, se CodeCatalyst actualiza el código fuente.

Cada proyecto realiza un seguimiento de la actividad del proyecto como una lista de eventos por usuario, por ejemplo, cuando se crea un proyecto o se modifica un recurso. La actividad del proyecto se supervisa y se agrega a nivel de espacio. Para obtener más información sobre cómo trabajar con datos de actividad, consulte [Ver todos los proyectos](#).

Si tu proyecto usa AWS recursos, puedes conectar tu CodeCatalyst cuenta a una AWS cuenta en la que tengas permisos administrativos para integrar los recursos del proyecto.

Puedes añadir repositorios de origen, ediciones y otros recursos a tu proyecto después de crearlo. Debe tener el rol de administrador de Space para crear proyectos.

Crear un proyecto en Amazon CodeCatalyst

Con CodeCatalyst los proyectos, puedes realizar tareas de desarrollo con flujos de trabajo y repositorios compartidos de integración continua o entrega continua (CI/CD), gestionar los recursos, realizar un seguimiento de los problemas y añadir usuarios.

Antes de crear un proyecto, debe tener el rol de administrador de Space o de usuario avanzado.

Temas

- [Crear un proyecto con un plano](#)
- [Crear un proyecto vacío en Amazon CodeCatalyst](#)

- [Crear un proyecto con un GitHub repositorio vinculado](#)

Crear un proyecto con un plano

Puedes aprovisionar todos los recursos del proyecto y el código de muestra con un plano de proyecto. Para obtener información sobre los planos, consulte la referencia del plan en. [Referencia del plano del proyecto](#)

Para crear un proyecto con un plano

1. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un proyecto.
2. En el panel de control del espacio, seleccione Crear proyecto.
3. Seleccione Empezar con un plano.
4. Elige un plano y, a continuación, selecciona Siguiente.
5. En Asigne un nombre al proyecto, introduzca el nombre que desee asignar al proyecto y los nombres de los recursos asociados. El nombre debe ser único dentro de su espacio.
6. En Recursos del proyecto, configure los parámetros comunes del proyecto.
7. (Opcional) Para ver los archivos de definición con actualizaciones basadas en las selecciones de parámetros del proyecto que haya realizado, seleccione Ver código o Ver flujo de trabajo en Generar vista previa del proyecto.
8. (Opcional) Seleccione Ver detalles en la tarjeta del blueprint para ver detalles específicos del blueprint, como una descripción general de la arquitectura del blueprint, las conexiones y los permisos necesarios y el tipo de recursos que crea el blueprint.
9. Elija Crear proyecto.

Para obtener más información sobre los esquemas de los proyectos, consulte. [Referencia del plano del proyecto](#)

Crear un proyecto vacío en Amazon CodeCatalyst

Puede crear un proyecto vacío sin recursos y añadir manualmente los recursos que desee más adelante.

Antes de crear un proyecto, debe tener el rol de administrador del espacio o de usuario avanzado.

Para crear un proyecto vacío

1. Vaya al espacio en el que desea crear un proyecto.
2. En el panel de control del espacio, seleccione Crear proyecto.
3. Elija Empezar desde cero.
4. En Asigne un nombre a su proyecto, introduzca el nombre que desea asignar a su proyecto. El nombre debe ser único dentro de su espacio.
5. Elija Crear proyecto.

Crear un proyecto con un GitHub repositorio vinculado

Puedes crear un CodeCatalyst proyecto nuevo que se vincule a un repositorio GitHub fuente. A continuación, puedes usar el repositorio de GitHub fuentes enlazado en tu CodeCatalyst proyecto.

Antes de crear un CodeCatalyst proyecto, debe tener el rol de administrador de Space o de usuario avanzado. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#) y [Invitar a un usuario directamente a un espacio](#).

Ya debe tener una GitHub cuenta y debe haber creado ya el repositorio al que desea vincular.

Para crear un proyecto CodeCatalyst que enlace a un repositorio fuente de tu GitHub cuenta, tendrás que completar las tres tareas siguientes:

1. Instala la extensión de GitHub repositorios.
2. Conecta tu GitHub cuenta a CodeCatalyst.
3. Crea un CodeCatalyst proyecto vinculado a tu GitHub cuenta.


Para instalar la extensión de GitHub repositorios

1. Vaya al espacio en el que desea crear un proyecto.
2. En el panel de control del espacio, seleccione Crear proyecto.
3. Selecciona Traiga su propio código.

Si la extensión de GitHub repositorios aún no está instalada, aparecerá un mensaje de instalación.

4. Elija Instalar. Revise los permisos que requiere la extensión y, si quiere continuar, seleccione Instalar de nuevo.

Tras instalar la extensión de GitHub repositorios, el siguiente paso es conectar la GitHub cuenta CodeCatalyst al espacio.

 Note

No puedes usar GitHub repositorios vacíos o archivados con los proyectos. CodeCatalyst La extensión de GitHub repositorios no es compatible con los repositorios de GitHub Enterprise Server.

Para conectar tu cuenta a GitHub CodeCatalyst

1. En la página Crear proyecto de CodeCatalyst, si no hay ninguna GitHub cuenta conectada, aparecerá un mensaje. Elija Connect GitHub account para ir al sitio externo GitHub.
2. Inicia sesión en tu GitHub cuenta con tus GitHub credenciales y, a continuación, elige la cuenta en la que quieres instalar Amazon CodeCatalyst.

 Tip

Si ya has conectado una GitHub cuenta al espacio, no se te pedirá que vuelvas a autorizar. En su lugar, aparecerá un cuadro de diálogo que le preguntará dónde desea instalar la extensión si es miembro o colaborador en más de un GitHub espacio, o la página de configuración de la CodeCatalyst aplicación Amazon si solo pertenece a un GitHub espacio. Configure la aplicación para el acceso al repositorio que desee permitir y, a continuación, seleccione Guardar. Si el botón Guardar no está activo, realice un cambio en la configuración e inténtelo de nuevo.

3. Elige si quieres permitir el acceso CodeCatalyst a todos los repositorios actuales y futuros, o elige los GitHub repositorios específicos en los que quieres usarlos. CodeCatalyst La opción predeterminada es incluir todos los GitHub repositorios de la GitHub cuenta, incluidos los repositorios futuros a los que accederá. CodeCatalyst
4. Revisa los permisos otorgados a y CodeCatalyst, a continuación, selecciona Instalar.

Tras conectar tu GitHub cuenta a CodeCatalyst, podrás vincular los GitHub repositorios de esa cuenta a tus CodeCatalyst proyectos.

Para crear su proyecto

1. En la página Crear proyecto, en el menú desplegable de la GitHub cuenta, realiza una de las siguientes acciones:
 - Elige una GitHub cuenta a la que ya te hayas conectado CodeCatalyst.
 - (Opcional) Si no ves la GitHub cuenta que quieres usar, selecciona Conectar una GitHub cuenta para ir a la página de extensiones CodeCatalyst. Para obtener más información, consulte [Uso de GitHub repositorios en CodeCatalyst](#).
2. En el menú desplegable del GitHub repositorio, los GitHub repositorios de la GitHub cuenta conectada se muestran en el menú desplegable. Elige el GitHub repositorio que quieres vincular a tu proyecto.
3. En el campo de entrada de texto Asigne un nombre a su proyecto, introduzca el nombre que desee asignar a su proyecto. El nombre debe ser único dentro de su espacio.
4. Elija Crear proyecto.

Una vez que el proyecto esté listo, puede añadir recursos y tareas.

- Para obtener más información sobre los flujos de trabajo de CI/CD creados con su proyecto, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#)
- Para trabajar con acciones de compilación similares a las de su nuevo proyecto que implementan artefactos de compilación en un bucket de Amazon S3, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#) y [Tutorial: Carga artefactos a Amazon S3](#).
- Para empezar con un proyecto vacío y trabajar en la implementación de una aplicación sin servidor similar con una implementación en AWS CloudFormation pila, consulte [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#).
- Para añadir un panel de planificación de problemas, consulte [Problemas en CodeCatalyst](#).
- Para ver la descripción general del proyecto, el estado del proyecto, la actividad reciente del equipo y el trabajo asignado, consulte [Visualización de un proyecto](#).
- Para ver el código fuente o crear una solicitud de incorporación de cambios, consulte [Repositorios de fuentes en CodeCatalyst](#).
- Para configurar notificaciones que envíen alertas de estado en caso de que el flujo de trabajo se ejecute correctamente o no, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).
- Para invitar a miembros a su proyecto, consulte [Gestión de los miembros del proyecto](#).
- Para configurar entornos de desarrollo, consulte [Entornos de desarrollo en CodeCatalyst](#).

Visualización de un proyecto

Desde tu CodeCatalyst espacio, puedes ver los detalles de cada proyecto en el que tengas permisos de proyecto.

Para ver un proyecto, debe ser miembro del proyecto o tener el rol de administrador del espacio.

Si aún no ha creado un proyecto, consulte [Crear un proyecto en Amazon CodeCatalyst](#). Debe tener el rol de administrador del espacio en el que desee crear un proyecto.

- En la descripción general del proyecto, puedes ver los miembros del proyecto, los repositorios de origen, los flujos de trabajo ejecutados, las solicitudes de cambios pendientes, los entornos de desarrollo del proyecto y los problemas.
- En la configuración del proyecto, puedes ver y gestionar los detalles del proyecto, eliminarlo, invitar a nuevos miembros al proyecto, gestionar los miembros del proyecto y configurar las notificaciones.

Visualización de las tareas del proyecto y los entornos de desarrollo

Para ver un resumen de las tareas del proyecto, como las emisiones pendientes y las solicitudes de cambios que te hayas asignado o creado, y los entornos de desarrollo asociados al proyecto, usa la consola.

Para ver un proyecto, debes ser miembro del proyecto o tener el rol de administrador del espacio.

Para ver los repositorios de origen, los flujos de trabajo ejecutados, las incidencias, las solicitudes de incorporación de datos, los entornos de desarrollo y los problemas

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, elija Overview (Información general).
4. Vea las tareas del proyecto que tiene asignadas y creadas por usted.
 - Ver la lista de miembros + Ver todos para ver una lista de los miembros del proyecto.
 - Consulta la tarjeta de repositorios para ver los repositorios de origen que están asociados al proyecto.

- Consulte la tarjeta Ejecuciones del flujo de trabajo para ver los flujos de trabajo asociados al proyecto.
- Consulta la tarjeta Abrir solicitudes de extracción para ver un resumen del estado del repositorio de código, además de las solicitudes de extracción que te hayan asignado y creado tú.
- Consulta la tarjeta Mis entornos de desarrollo para ver un resumen de los entornos de desarrollo asociados al proyecto.
- Consulta la tarjeta Problemas para ver un resumen de las tareas asignadas o de las tareas que has creado.

Ver todos los proyectos

En la lista de proyectos de tu espacio, puedes ver todos los proyectos en los que tienes permisos.

Para ver un resumen de las tareas del proyecto, como las emisiones pendientes y las solicitudes de cambios que te hayas asignado o creado, y los entornos de desarrollo asociados al proyecto, usa la consola.

Para ver un proyecto, debes ser miembro del proyecto o tener el rol de administrador del espacio.

Para ver los repositorios de origen, los flujos de trabajo ejecutados, las incidencias, las solicitudes de incorporación de datos, los entornos de desarrollo y los problemas

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Vea el nombre del proyecto, la ruta, el ID del proyecto y la descripción.

Visualización de la configuración del proyecto

En la configuración del proyecto, puedes ver los miembros del proyecto, los repositorios de origen, los flujos de trabajo ejecutados, las solicitudes de cambios pendientes, los entornos de desarrollo del proyecto y los problemas.

Para ver un resumen de las tareas del proyecto, como las emisiones pendientes y las solicitudes de extracción que te hayas asignado o creado, y los entornos de desarrollo asociados al proyecto, utiliza la consola.

Para ver tus repositorios de origen, los flujos de trabajo ejecutados, las incidencias, las solicitudes de cambios, los entornos de desarrollo y las incidencias

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Vea el nombre del proyecto, la ruta, el ID del proyecto y la descripción.

Cambiar a un proyecto diferente en CodeCatalyst

Para cambiar a un proyecto diferente, usa la consola para elegir de una lista de proyectos a los que tienes acceso.

Para cambiar a un proyecto diferente

1. En la CodeCatalyst consola, selecciona el selector de proyectos de la parte superior.
2. Expande el menú desplegable y elige el proyecto al que deseas navegar.

Eliminar un proyecto en Amazon CodeCatalyst

Puedes eliminar un proyecto para eliminar todos los accesos a los recursos del proyecto. Debe tener el rol de administrador del espacio o administrador del proyecto para eliminar un proyecto. Una vez que hayas eliminado un proyecto, los miembros del proyecto no podrán acceder a los recursos del proyecto y se detendrán todos los flujos de trabajo que soliciten repositorios de fuentes de terceros.

Para eliminar tu proyecto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Elija Delete project (Eliminar proyecto).

5. Introduzca **delete** para confirmar la eliminación.
6. Elija Delete project (Eliminar proyecto).

Gestión de los miembros del proyecto

Puedes gestionar los miembros de tus proyectos mediante la CodeCatalyst consola de Amazon. Puedes añadir o eliminar usuarios, gestionar los roles de los miembros actuales, enviar invitaciones para unirse a tu proyecto y cancelar las invitaciones que aún no se hayan aceptado.

En la página de miembros para los usuarios del espacio y del proyecto, los usuarios pueden tener varios roles. Los usuarios con varios roles mostrarán un indicador cuando tengan varios roles y aparecerán primero con el rol con más permisos.

Ver los miembros de un proyecto

Cuando agregas un usuario a tu proyecto, asignas un rol que otorga los permisos del proyecto de la siguiente manera:

- El rol de administrador del proyecto tiene todos los permisos de un proyecto. Asigne esta función únicamente a los usuarios que necesiten administrar todos los aspectos de un proyecto, incluida la edición de la configuración del proyecto, la administración de los permisos del proyecto y la eliminación del proyecto. Para obtener más información, consulte [Función de administrador del proyecto](#).
- El rol de colaborador tiene los permisos necesarios para trabajar en un proyecto. Asigna este rol a los usuarios que necesiten trabajar con el código, los flujos de trabajo, los problemas y las acciones de un proyecto. Para obtener más información, consulte [Función de colaborador](#).
- El rol de revisor tiene permisos de revisión. Para obtener más detalles, consulte [Función de revisor](#).
- La función de solo lectura tiene permisos de lectura. Para obtener más detalles, consulte [Función de solo lectura](#).

No es necesario que invite a un usuario con la función de administrador del espacio a su proyecto, ya que este ya tiene acceso implícito a todos los proyectos del espacio.

Al invitar a un usuario a su proyecto (sin asignar la función de administrador del espacio), el usuario aparecerá en la tabla de miembros del proyecto, en los proyectos, y en la tabla de miembros del proyecto, en los espacios.

Para ver los usuarios y los roles de tu espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Seleccione la pestaña Miembros.

La tabla de miembros del proyecto muestra todos los miembros que tienen un rol en un proyecto.

Tip

Si tiene el rol de administrador del espacio, puede ver los proyectos a los que ha sido invitado directamente. Ve a la configuración del proyecto y, a continuación, selecciona Mis proyectos.

En la tabla de administradores del espacio se muestran los usuarios con el rol de administrador del espacio. Estos usuarios se asignan automáticamente (de forma implícita) a todos los proyectos del espacio y no tienen ningún rol en ningún proyecto.

En la columna Estado, los siguientes son valores válidos:

- CodeCatalyst Invitado: se envió la invitación, pero el usuario aún no la ha aceptado o rechazado.
- Miembro: el usuario ha aceptado la invitación.

Temas

- [Invitar a un usuario a tu proyecto](#)
- [Cancelar una invitación](#)
- [Eliminar un usuario de tu proyecto](#)
- [Aceptar o rechazar una invitación para un proyecto](#)

Invitar a un usuario a tu proyecto

Puedes usar la consola para invitar a usuarios a tu proyecto. Puedes invitar a miembros de tu espacio o añadir nombres ajenos a tu espacio.

Para invitar a usuarios a tu proyecto, debes iniciar sesión con el rol de administrador del proyecto o administrador del espacio.

No es necesario que invite a un usuario con el rol de administrador del espacio a su proyecto, ya que este ya tiene acceso implícito a todos los proyectos del espacio.

Al invitar a un usuario a su proyecto (sin asignar la función de administrador del espacio), el usuario aparecerá en la tabla de miembros del proyecto, en los proyectos, y en la tabla de miembros del proyecto, en los espacios.

Para invitar a un miembro a tu proyecto desde la pestaña de configuración del proyecto

1. Vaya a su proyecto.

Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.

2. En el panel de navegación, selecciona Configuración del proyecto.
3. Seleccione la pestaña Miembros.
4. En Miembros del proyecto, elija Invitar a un nuevo miembro.
5. Escribe la dirección de correo electrónico del nuevo miembro, elige el rol de este miembro y, a continuación, selecciona Invitar. Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Para invitar a un miembro a tu proyecto desde la página de resumen del proyecto

1. Vaya a su proyecto.

Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.

2. Pulse el botón Members +.

3. Escriba la dirección de correo electrónico del nuevo miembro, elija el rol de este miembro y, a continuación, elija Invitar. Para obtener más información acerca de los roles de , consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Cancelar una invitación

Si has enviado una invitación recientemente, puedes cancelarla siempre y cuando aún no se haya aceptado.

Para gestionar las invitaciones a proyectos, debes tener el rol de administrador del proyecto o administrador del espacio.

Para cancelar la invitación de un miembro del proyecto

1. Navega hasta el proyecto al que has enviado la invitación que deseas cancelar.
2. En el panel de navegación, selecciona Configuración del proyecto.
3. Consulte la pestaña Miembros y compruebe que el miembro tiene el estado Invitado.

Note

Solo puedes cancelar una invitación que aún no haya sido aceptada.

4. Elige la opción situada junto a la fila con el miembro invitado y, a continuación, selecciona Cancelar la invitación.
5. Aparece una ventana de confirmación. Selecciona Cancelar la invitación para confirmarla.

Eliminar un usuario de tu proyecto

Puedes usar la consola para eliminar un usuario de tu proyecto.

Para eliminar un usuario del proyecto, debe iniciar sesión con el rol de administrador del proyecto o administrador del espacio.

Note

Al eliminar a un usuario de todos los proyectos de un espacio, se elimina automáticamente al usuario de ese espacio.

Para eliminar un usuario de un proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio donde se encuentra el proyecto que desea ver. En Proyectos, elige tu proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Seleccione la pestaña Miembros.
5. Seleccione el selector situado junto al perfil que desee eliminar y, a continuación, seleccione Eliminar.
6. Confirme que desea eliminar al usuario y, a continuación, seleccione Eliminar.

Aceptar o rechazar una invitación para un proyecto

Es posible que recibas una invitación por correo electrónico para unirse a un CodeCatalyst proyecto de Amazon. Puede aceptar o rechazar la invitación.

Para aceptar o rechazar una invitación

1. Abre el correo electrónico de invitación.
2. Elige el enlace del proyecto en el correo electrónico.
3. Selecciona Aceptar o Rechazar.

Si eliges Rechazar, se enviará un correo electrónico a la cuenta de gestión del proyecto para notificarle que has rechazado la invitación.

Gestión de equipos para proyectos

Después de crear un proyecto, puedes añadir equipos. Los equipos te permiten agrupar a los usuarios para que puedan compartir los permisos y gestionar los proyectos, el seguimiento de los problemas, las funciones y los recursos CodeCatalyst como miembros del proyecto y del espacio.

Debes tener el rol de administrador de proyectos para gestionar los equipos de tu proyecto.

Temas

- [Añadir un equipo a un proyecto](#)
- [Administrar un rol de proyecto para un equipo](#)

- [Eliminar un rol de proyecto para un equipo](#)

Añadir un equipo a un proyecto

Puedes gestionar equipos donde los miembros del equipo puedan acceder a los recursos de tu proyecto.

En la página de miembros para los usuarios del espacio y del proyecto, los usuarios pueden tener varios roles. Los usuarios con varios roles mostrarán un indicador cuando tengan varios roles y aparecerán primero con el rol con más permisos.

Para añadir un equipo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su proyecto. Elija la configuración del proyecto y, a continuación, elija Teams.
3. Selecciona Añadir equipo.
4. En Equipo, elige un equipo de la lista de equipos disponibles.
5. En Función de proyecto, elige una función de la lista de funciones de proyecto disponibles en CodeCatalyst.
 - Administrador del proyecto: para obtener más información, consulte [Función de administrador del proyecto](#).
 - Colaborador: para obtener más información, consulte [Función de colaborador](#).
 - Revisor: para obtener más información, consulte [Función de revisor](#).
 - Solo lectura: para obtener más información, consulte [Función de solo lectura](#).
6. Selecciona Añadir equipo.

Administrar un rol de proyecto para un equipo

Un equipo puede tener permisos de rol, como el de usuario avanzado, en un espacio. Puedes cambiar el rol de un equipo en el espacio, pero ten en cuenta que todos los miembros del equipo heredarán esos permisos.

Para añadir o cambiar un rol de proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. Vaya a su espacio. Elija la configuración del proyecto y, a continuación, elija Teams.
3. Para cambiar un rol, selecciona el selector situado junto al equipo en esta lista y, a continuación, selecciona Cambiar rol. Para añadir un rol, selecciona Añadir rol de proyecto. En Proyecto, elija el proyecto que desee añadir y, en Función, elija la función. Elige uno de los roles de proyecto disponibles:
 - Administrador del proyecto: para obtener más información, consulte [Función de administrador del proyecto](#).
 - Colaborador: para obtener más información, consulte [Función de colaborador](#).
 - Revisor: para obtener más información, consulte [Función de revisor](#).
 - Solo lectura: para obtener más información, consulte [Función de solo lectura](#).
4. Seleccione Guardar.

Eliminar un rol de proyecto para un equipo

En CodeCatalyst, puedes ver las funciones del proyecto para tu equipo. También puedes ver los miembros de un equipo. Puedes eliminar el rol de proyecto de un equipo.

Para eliminar un rol de proyecto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su espacio. Elija la configuración del proyecto y, a continuación, elija Teams.
3. Selecciona la pestaña Funciones del proyecto.
4. Elige el rol que deseas eliminar.

Important

Al eliminar un rol de un equipo, se eliminan los permisos asociados para todos los usuarios del equipo.

5. Elija Guardar.

Gestión de los recursos de la máquina

Un recurso de máquina representa tu identidad con respecto al recurso autorizado cuando accedes a CodeCatalyst través del inicio de sesión único. Los recursos de la máquina se utilizan para conceder

permisos a los recursos del proyecto, como los planos y los flujos de trabajo. Puede ver los recursos de máquina de su proyecto y puede elegir habilitar o deshabilitar los recursos de máquina para su proyecto. Por ejemplo, es posible que desee deshabilitar un recurso de máquina para administrar el acceso y volver a habilitarlo más adelante.

Estas operaciones están disponibles para los recursos de máquina en los casos en que sea necesario revocar o deshabilitar un recurso de máquina. Por ejemplo, si sospecha que las credenciales podrían estar comprometidas, puede deshabilitar el recurso de la máquina. Por lo general, no será necesario utilizar estas operaciones.

Debe tener el rol de administrador de espacios o el rol de administrador de proyectos para ver esta página y administrar los recursos de la máquina a nivel de proyecto.

Temas

- [Ver los recursos de la máquina](#)
- [Desactive los recursos de la máquina](#)
- [Habilite los recursos de la máquina](#)

Ver los recursos de la máquina

Puede ver una lista de los recursos de la máquina que se utilizan en su proyecto.

Debe tener el rol de administrador del espacio o el rol de administrador del proyecto.

Para ver los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto y, a continuación, seleccione la configuración del proyecto. Elige Recursos de la máquina.
3. En el menú desplegable, elija Acción de flujo de trabajo para ver solo los recursos de máquina para los flujos de trabajo. Elija Blueprint para ver solo los recursos de máquina para los blueprints.

También puede filtrar por nombre mediante el campo Filtro.

Desactive los recursos de la máquina

Puede optar por deshabilitar los recursos de la máquina que se utilizan en su proyecto.

⚠ Important

Al deshabilitar los recursos de la máquina, se eliminarán todos los permisos de todos los blueprints o flujos de trabajo asociados en el espacio.

Debe tener el rol de administrador del espacio o el rol de administrador del proyecto.

Para deshabilitar los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto y, a continuación, seleccione la configuración del proyecto. Elige Recursos de la máquina.
3. Elija una de las siguientes opciones.

⚠ Important

Al deshabilitar los recursos de la máquina, se eliminarán todos los permisos de todos los blueprints o flujos de trabajo asociados en el espacio.

- Para deshabilitar uno por uno, elija el selector situado junto a uno o más recursos de máquina que desee deshabilitar. Elija Desactivar y, a continuación, elija Este recurso.
- Para deshabilitar todos los recursos, selecciona Desactivar y, a continuación, selecciona Todos los recursos.
- Para deshabilitar todas las acciones del flujo de trabajo, selecciona Desactivar y, a continuación, selecciona Todas las acciones del flujo de trabajo.
- Para deshabilitar todos los esquemas, seleccione Desactivar y, a continuación, elija Todos los esquemas.

Habilite los recursos de la máquina

Puede optar por habilitar los recursos de máquina que están en uso en su proyecto y que se han deshabilitado.

Debe tener el rol de administrador del espacio o el rol de administrador del proyecto.

Para habilitar los recursos de la máquina

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto y, a continuación, seleccione la configuración del proyecto. Elige Recursos de la máquina.
3. Elija una de las siguientes opciones.
 - Para activarlos de forma individual, seleccione el selector situado junto a uno o más recursos de máquina que desee activar. Elija Habilitar y, a continuación, elija Este recurso.
 - Para habilitar todos los recursos, elija Habilitar y, a continuación, elija Todos los recursos.
 - Para habilitar todas las acciones del flujo de trabajo, elija Habilitar y, a continuación, elija Todas las acciones del flujo de trabajo.
 - Para habilitar todos los esquemas, seleccione Habilitar y, a continuación, elija Todos los esquemas.

Cuotas para proyectos en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de los proyectos en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Número máximo de proyectos por espacio	100
Número máximo de proyectos a los que puede pertenecer un usuario	1 000
Número máximo de miembros que pueden pertenecer a un proyecto.	10 000
Nombres de proyectos	Los nombres de los proyectos deben ser únicos dentro de un espacio. Los nombres deben tener entre 3 y 63 caracteres. Los nombres distinguen mayúsculas de minúsculas. Los nombres de los proyectos deben comenzar con un carácter alfanumérico. Caracteres válidos: A-Z, a-z, 0-9, espacios y, _ (guión bajo) - (guión)

Los nombres de los proyectos no pueden contener ninguno de los siguientes caracteres:

! ? @ # \$ % ^ & * () + = { } [] | \ / > < ~ ` ' " ; :

Descripciones de proyectos

Las descripciones de los proyectos pueden tener un máximo de 200 caracteres. Caracteres válidos: A-Z, a-z, 0-9, espacios y., _ (guión bajo) - (guión). Las descripciones de proyectos son opcionales.

Trabajar con notificaciones en CodeCatalyst

Puede configurar notificaciones para supervisar sus proyectos y recursos CodeCatalyst. Los usuarios pueden elegir los eventos del proyecto sobre los que desean recibir correos electrónicos en cualquier proyecto del que sean miembros. También puedes configurar las notificaciones que se envían a todo un equipo en una aplicación de mensajería para equipos, como Slack, configurando el acceso entre un CodeCatalyst espacio y un espacio de trabajo de Slack y, a continuación, configurando las notificaciones de un proyecto para que se envíen a uno o más canales de ese espacio de trabajo de Slack. Una vez que hayas configurado el acceso entre un CodeCatalyst espacio y un espacio de trabajo de Slack, los miembros del proyecto también tendrán la opción de añadir sus propios ID de miembro de Slack para que puedan recibir notificaciones directas sobre CodeCatalyst los eventos en los espacios de trabajo y canales de Slack conectados.

Note

El conjunto de eventos del proyecto que se puede enviar a Slack no es el mismo conjunto de eventos sobre los que los usuarios pueden elegir que se les notifique por correo electrónico.

Temas

- [¿Cómo funcionan las notificaciones?](#)
- [Cómo empezar a usar las notificaciones de Slack](#)
- [Administrar las notificaciones en Amazon CodeCatalyst](#)

¿Cómo funcionan las notificaciones?

Puedes configurar tu proyecto para que envíe notificaciones a la aplicación de mensajería de tu equipo, como Slack.

¿Qué permisos son necesarios para las notificaciones?

Cualquier miembro del proyecto puede configurar, ver, actualizar o eliminar los ajustes de notificación de un canal en CodeCatalyst. Sin embargo, solo los usuarios con el rol de administrador de espacios pueden añadir o eliminar espacios de trabajo de Slack. Todos los usuarios pueden configurar los eventos de proyecto sobre los que quieren recibir correos electrónicos para los proyectos a los que pertenecen. CodeCatalyst

¿Sobre qué CodeCatalyst eventos puedo configurar las notificaciones?

Puedes configurarlo CodeCatalyst para enviar notificaciones a uno o más canales de Slack sobre los eventos del flujo de trabajo. Una vez configuradas las notificaciones entre un CodeCatalyst proyecto y Slack, los usuarios del proyecto pueden optar por añadir su propio ID de miembro de Slack para recibir mensajes directos sobre los eventos en los canales de Slack. CodeCatalyst Los usuarios que añadan sus seudónimos de miembro de Slack recibirán menciones directas en sus identificadores en los canales de Slack configurados para sus proyectos, lo que ayudará a dar a conocer los eventos que les interesan.

También puedes elegir los eventos sobre los que quieres recibir correos electrónicos. Estos correos electrónicos se envían a la dirección de correo electrónico configurada para tu ID de AWS constructor.

¿Cómo aparecen las notificaciones?

Puedes configurarlo CodeCatalyst para enviar notificaciones a uno o más canales de Slack. Debes autorizarte a conceder permisos CodeCatalyst para acceder a tu espacio de trabajo de Slack. Una vez otorgada la autorización, CodeCatalyst podrás enviar notificaciones a los canales de Slack que configures. Si un miembro del proyecto decide añadir su ID de miembro de Slack, podrá recibir menciones sobre CodeCatalyst los eventos en los canales de Slack configurados para ese proyecto.


¿Cómo configuro las notificaciones?

Las notificaciones por correo electrónico se configuran como parte de CodeCatalyst. Los usuarios del proyecto pueden elegir los eventos sobre los que desean recibir correos electrónicos en su página de Mi configuración.

Para configurar las notificaciones de Slack para los recursos de tu proyecto, debes completar las siguientes tareas de alto nivel.

Para configurar las notificaciones (tareas de alto nivel)

1. En CodeCatalyst, configuras una conexión entre CodeCatalyst y un cliente de mensajería, como Slack. Una vez conectado un espacio de trabajo de Slack, estará disponible para todos los proyectos del espacio.

 Note

Solo los usuarios con un rol de administrador de espacios pueden añadir o eliminar un espacio de trabajo de Slack.

2. En tu proyecto CodeCatalyst, añade el canal en el que quieres que tu equipo reciba las notificaciones.
3. En CodeCatalyst, activas las notificaciones de varios eventos, como un fallo en la ejecución del flujo de trabajo, y especificas el canal al que quieres que se envíen.


Para conocer los pasos en detalle, consulte [Cómo empezar a usar las notificaciones de Slack](#).

Una vez configuradas las notificaciones entre un CodeCatalyst espacio y Slack, los usuarios pueden optar por añadir sus propios identificadores de miembros de Slack para recibir mensajes directos sobre los CodeCatalyst eventos en los canales de Slack configurados para sus proyectos.

Cómo empezar a usar las notificaciones de Slack

Después de crear un proyecto, puedes configurar las notificaciones de Slack que ayuden a tu equipo a supervisar los recursos del proyecto.

Estos pasos te explican cómo configurar las notificaciones de Slack por primera vez. CodeCatalyst Si ya has configurado las notificaciones, consulta [Administrar las notificaciones en Amazon CodeCatalyst](#).

 Note

El conjunto de eventos del proyecto que se puede enviar a los canales de notificación no es el mismo conjunto de eventos sobre los que los usuarios pueden elegir que se les notifique

por correo electrónico. Para obtener más información, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Conéctate CodeCatalyst a tu espacio de trabajo de Slack](#)
- [Paso 2: Añade tu canal de Slack a CodeCatalyst](#)
- [Paso 3: Prueba las notificaciones desde CodeCatalyst Slack](#)
- [Paso 4: Sigüentes pasos](#)

Requisitos previos

Antes de comenzar, necesitará lo siguiente:

- Un CodeCatalyst espacio. Para obtener información sobre cómo crear un CodeCatalyst espacio e iniciar sesión por primera vez, consulte [Configuración CodeCatalyst](#).
- Un CodeCatalyst proyecto. Para obtener más información, consulte [Crear un proyecto en Amazon CodeCatalyst](#).
- Una CodeCatalyst cuenta con el rol de administrador del proyecto o administrador del espacio. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).
- Una cuenta de Slack y un espacio de trabajo de Slack a los que se pueda acceder mediante CodeCatalyst
- Un canal de Slack al que se CodeCatalyst enviarán las notificaciones. El canal puede ser público o privado.

Paso 1: Conéctate CodeCatalyst a tu espacio de trabajo de Slack

Solo los usuarios con el rol de administrador de espacios pueden añadir o eliminar espacios de trabajo de Slack. Añadir o eliminar un espacio de trabajo de Slack afecta a todos los proyectos del espacio. Para establecer la conexión entre CodeCatalyst y Slack, CodeCatalyst realiza un protocolo de autenticación de OAuth seguro con tu espacio de trabajo de Slack.

Sigue las siguientes instrucciones para conectarte a tu espacio de trabajo de Slack CodeCatalyst .

Note

Esto solo debe hacerse una vez para cada espacio de trabajo de Slack. A continuación, puedes configurar las notificaciones por canal de Slack.

Para conectarte CodeCatalyst a tu espacio de trabajo de Slack

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Seleccione la pestaña Notificaciones.
5. Seleccione Configurar notificaciones.
6. Elige Conectar al espacio de trabajo de Slack.
7. Lee el contenido del cuadro de diálogo y, a continuación, selecciona Conectar al espacio de trabajo de Slack.
8. En el mensaje del AWSChatbot:
 - a. En la esquina superior derecha, elige el espacio de trabajo de Slack que contiene tu canal.
 - b. Elija Allow.

Volverás a la CodeCatalyst consola.

9. Siga en [Paso 2: Añade tu canal de Slack a CodeCatalyst](#).

Paso 2: Añade tu canal de Slack a CodeCatalyst

Necesitas el ID del canal de Slack para añadir tu canal. CodeCatalyst

Para obtener el ID de tu canal de Slack

1. Inicia sesión en Slack. Para obtener más información, consulta [Iniciar sesión en Slack](#).
2. Ve al espacio de trabajo de Slack que contiene el canal al que quieres que vayan las notificaciones. Para obtener más información, consulta [Cambiar de un espacio de trabajo de Slack o iniciar sesión en otros espacios](#) de trabajo de Slack.

3. En el panel de navegación, abre el menú contextual (haz clic con el botón derecho) del canal al que quieres que vayan las notificaciones y selecciona Abrir los detalles del canal.

El ID del canal aparece en la parte inferior del cuadro de diálogo.

4. Copie el valor del ID del canal. Lo necesitará en el siguiente paso.

Ahora puedes conectar tu canal de Slack con el ID de canal que acabas de copiar. CodeCatalyst

Para añadir tu canal de Slack a CodeCatalyst

1. Antes de empezar, si tu canal de Slack es privado, añade la aplicación AWS Chatbot al canal de la siguiente manera:
 - a. En el cuadro de mensajes de tu canal de Slack, introduce **@aws** y selecciona una aplicación en el cuadro de diálogo.
 - b. Pulse Intro.

Aparece un mensaje de Slackbot que indica que AWS Chatbot no está en el canal privado.
 - c. Selecciona Invitarlos para invitar al AWS Chatbot al canal.
2. En la CodeCatalyst consola, selecciona Siguiente.
3. En el ID del canal, pega el ID del canal de Slack que obtuviste anteriormente.
4. En Nombre del canal, introduce un nombre. Te recomendamos usar el nombre del canal de Slack.
5. Elija Siguiente.
6. En Seleccionar eventos de notificación, elige el tipo de evento del que quieres recibir notificaciones.
7. Elija Finalizar.

Paso 3: Prueba las notificaciones desde CodeCatalyst Slack

Una vez que tu proyecto esté configurado para enviar notificaciones sobre el estado del flujo de trabajo, podrás ver tus notificaciones en Slack.

Para ver tus notificaciones en Slack

1. En tu CodeCatalyst proyecto, [inicia un flujo de trabajo manualmente](#) para completar una ejecución del flujo de trabajo y recibir una notificación de estado cuando finalice la ejecución.
2. En Slack, consulta el canal que configuraste para las notificaciones. Tus notificaciones muestran el estado más reciente de cada flujo de trabajo ejecutado y si se ha ejecutado correctamente o no.

Paso 4: Sigüientes pasos

Una vez que hayas configurado un espacio de trabajo de Slack para tu CodeCatalyst espacio, podrás añadir canales de Slack adicionales a los CodeCatalyst proyectos existentes y añadirlos a los proyectos nuevos una vez que los hayas creado. También puedes informar a los usuarios del proyecto de que pueden configurar las notificaciones personales de Slack para sus ID de miembro de Slack y configurar los eventos para los que recibirán correos electrónicos. Para obtener más información, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).

Administrar las notificaciones en Amazon CodeCatalyst

Puedes configurarlo CodeCatalyst para enviar notificaciones sobre los eventos de tu proyecto. Puedes enviar notificaciones a clientes de mensajería, como los canales de Slack. Los usuarios del proyecto pueden elegir los eventos del proyecto sobre los que se les notificará mediante correos electrónicos enviados a la dirección de correo electrónico configurada para su perfil.

Note

El conjunto de eventos del proyecto que se pueden enviar a los canales de notificación no es el mismo conjunto de eventos sobre los que los usuarios pueden elegir que se les notifique por correo electrónico.

Temas

- [Administrar las notificaciones que se te envían directamente](#)
- [Administrar las notificaciones enviadas a los canales](#)

Administrar las notificaciones que se te envían directamente

Puedes elegir que te envíen notificaciones por correo electrónico sobre los eventos de cualquier proyecto del que seas miembro. Estos correos electrónicos se enviarán a la dirección de correo electrónico configurada en tu ID de AWS constructor. De forma predeterminada, recibirá correos electrónicos sobre todos los eventos del proyecto para los que se puedan enviar correos electrónicos.

Si un proyecto se ha configurado para enviar notificaciones a un canal de Slack, puedes añadir tu ID de miembro de Slack para recibir menciones directas sobre CodeCatalyst los eventos de ese canal de Slack. Esto puede ayudarte a dar a conocer mejor los eventos que tienen lugar en los proyectos en los que desempeñas una función.

Para configurar las notificaciones por correo electrónico para los eventos del proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de menú superior, selecciona la insignia de tu perfil y, a continuación, selecciona Mi configuración. Se abre la página CodeCatalyst Mis ajustes.

Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

3. En Notificaciones por correo electrónico, busca el proyecto en la lista en la que quieres configurar las notificaciones por correo electrónico y selecciona Editar.
4. Selecciona los eventos para los que quieres recibir correos electrónicos y, a continuación, selecciona Guardar.

Para configurar las notificaciones personales de Slack

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de menú superior, selecciona la insignia de tu perfil y, a continuación, selecciona Mi configuración. Se abre la página CodeCatalyst Mis ajustes.

Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

3. En las notificaciones personales de Slack, selecciona Connect Slack ID y, a continuación, selecciona Conectar al espacio de trabajo de Slack. Se abrirá una ventana independiente.

Tip

Esta opción no se puede configurar a menos que un usuario con el rol de administrador del espacio haya añadido un espacio de trabajo de Slack a tu CodeCatalyst espacio. Para obtener más información, consulte [Cómo empezar a usar las notificaciones de Slack](#) y [Administrar las notificaciones enviadas a los canales](#).

4. En la ventana de solicitud de permisos, asegúrate de que el nombre del espacio de trabajo coincida con el espacio de trabajo de Slack configurado para el CodeCatalyst espacio. Selecciona Permitir para permitir el AWS Chatbot acceso al espacio de trabajo. La ventana se cerrará y el espacio de trabajo de Slack mostrará el estado de la conexión como Conectado.

Tip

Si el estado de la conexión no cambia, comprueba si se ha producido un error al conectar el espacio de trabajo de Slack. Puede que tengas que desplazarte hacia arriba para ver el error.

5. Para dejar de recibir notificaciones personales de Slack, selecciona el espacio de trabajo de Slack conectado y, a continuación, selecciona Desconectar el ID de Slack.

Administrar las notificaciones enviadas a los canales

Puedes añadir y gestionar las notificaciones sobre los eventos del proyecto si CodeCatalyst se envían a los recursos del equipo, como un canal de Slack del equipo. De este modo, puedes asegurarte de que todo tu equipo esté al tanto de los eventos importantes, como cuando se produce un error en la ejecución de un flujo de trabajo.

Note

Cualquier miembro de un proyecto puede gestionar las notificaciones enviadas a los canales de ese proyecto. Sin embargo, solo los usuarios con el rol de administrador de espacios pueden añadir o eliminar espacios de trabajo de Slack.

Añadir un canal de notificaciones para un proyecto

Puedes añadir un canal en el que quieras recibir notificaciones, como el canal de Slack de tu equipo.

Para añadir un canal de Slack para las notificaciones

1. Si vas a añadir tu primer canal de Slack, consulta en su lugar. [Cómo empezar a usar las notificaciones de Slack](#)

Tras configurar tu primer canal, vuelve a este procedimiento para configurar canales adicionales.

2. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navegue hasta su proyecto.
4. En el panel de navegación, selecciona Configuración del proyecto.
5. Seleccione la pestaña Notificaciones.
6. Elija Add channel (Agregar canal).
7. Seleccione Elegir espacio de trabajo y, a continuación, seleccione el espacio de trabajo de Slack que contiene el canal al que quieres enviar las notificaciones.

Si tu espacio de trabajo de Slack no está en la lista, puedes añadirlo siguiendo las instrucciones que aparecen en esta sección. [Cómo empezar a usar las notificaciones de Slack](#)

8. Antes de introducir un ID de canal, si el canal de Slack que quieres añadir es privado, sigue estos pasos:
 - a. En el cuadro de mensajes de tu canal de Slack, introduce **@aws** y selecciona una aplicación en la ventana emergente.
 - b. Pulse Intro.

Aparece un mensaje de Slackbot que indica que AWS Chatbot no está en el canal privado.

 - c. Seleccione Invitarlos para invitar al AWS Chatbot al canal.

9. En el campo ID CodeCatalyst del canal, introduce el ID del canal de Slack. Para encontrar el ID, ve a Slack y, en el panel de navegación, haz clic con el botón derecho en el canal y selecciona Abrir detalles del canal.

El ID del canal aparece en la parte inferior del cuadro de diálogo.

10. En Nombre del canal, introduzca un nombre. Te recomendamos que utilices el nombre del canal de Slack.
11. En Seleccionar eventos de notificación, elige el tipo de evento del que quieres recibir notificaciones.
12. Elija Add (Agregar).

Edición de las notificaciones de un canal de notificaciones

Puedes cambiar los canales a los que van las notificaciones y puedes desactivar notificaciones específicas por completo.

Para editar las notificaciones

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto.
3. En el panel de navegación, selecciona Configuración del proyecto.
4. Seleccione la pestaña Notificaciones.
5. Seleccione Editar notificaciones.
6. Haga una de las siguientes acciones:
 - Para enviar una notificación a un canal específico, selecciona el canal en la lista desplegable.
 - Para desactivar una notificación de forma global, selecciona el botón situado junto a la notificación.
 - Para dejar de enviar una notificación a un canal específico, selecciona la X del canal.
7. Seleccione Guardar.

Eliminar un canal

Para eliminar un canal

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. Navegue hasta su proyecto. En el panel de navegación, selecciona Configuración del proyecto.
3. En la página de configuración del proyecto, seleccione la pestaña Notificaciones.
4. Elige el indicador situado junto al canal que quieres eliminar y, a continuación, selecciona Eliminar canal. Cuando se te pida, selecciona Aceptar en la ventana de confirmación.

Planos en CodeCatalyst

Los planos son generadores de código arbitrario que representan un componente arquitectónico de un CodeCatalyst proyecto. El componente puede consistir en cualquier cosa, desde un flujo de trabajo en un solo archivo hasta todo el proyecto completo con un código de ejemplo. Los planos toman un conjunto arbitrario de opciones y las utilizan para generar un conjunto arbitrario de código de salida que se reenvía a un proyecto. A medida que el plan se actualiza con las mejores prácticas más recientes o las nuevas opciones, puede regenerar las partes relevantes de la base de código en los proyectos que contengan ese plan.

Puedes usar un CodeCatalyst plano de Amazon para crear un proyecto completo con un repositorio fuente, ejemplos de código fuente, flujos de trabajo de CI/CD, informes de compilación y prueba y herramientas integradas de seguimiento de problemas. Un CodeCatalyst plano genera recursos y código fuente en función de los parámetros de configuración establecidos. Cuando se utiliza un esquema CodeCatalyst gestionado, el modelo que elija determina los recursos que se añaden al proyecto, así como las herramientas que se CodeCatalyst crean o configuran, para que pueda realizar un seguimiento de los recursos del proyecto y utilizarlos. Como usuario de un blueprint, puede crear un proyecto con un blueprint o aplicarlo a un proyecto existente. CodeCatalyst Puede aplicar varios planos a su proyecto y cada uno de ellos se puede aplicar como un componente independiente. Por ejemplo, puede crear un proyecto con un blueprint de aplicación web y, posteriormente, aplicar un blueprint de seguridad. Cuando se actualiza uno de los planos, puede incorporar los cambios o correcciones en el proyecto mediante la gestión del ciclo de vida. Para obtener más información, consulte [Referencia del plano del proyecto](#) y [Trabajar con la administración del ciclo de vida como usuario de blueprint](#).

Como autor de planos, también puedes crear y publicar planos personalizados para que los miembros de tu CodeCatalyst espacio utilicen los recursos del proyecto. Los planos personalizados se pueden desarrollar para satisfacer las necesidades específicas de los proyectos de tu espacio. Tras añadir un plano personalizado al catálogo de planos de tu espacio, puedes gestionarlo y seguir realizando actualizaciones para que los proyectos de tu espacio estén actualizados con las mejores prácticas más recientes. Para obtener más información, consulte [Trabajar con planos personalizados en CodeCatalyst](#). [Para ver el SDK de planos y los ejemplos de planos, consulta el repositorio de código abierto. GitHub](#)

Temas

- [Crear un proyecto con un plano](#)
- [Aplicación y disociación de planos en un proyecto](#)

- [Actualización de un plano en un proyecto](#)
- [Edición de la descripción de un plano de un proyecto](#)
- [Trabajar con la administración del ciclo de vida como usuario de blueprint](#)
- [Referencia del plano del proyecto](#)
- [Trabajar con planos personalizados en CodeCatalyst](#)
- [Cuotas de planos en CodeCatalyst](#)

Crear un proyecto con un plano

Puedes crear rápidamente un proyecto utilizando un plano del catálogo de Amazon o el CodeCatalyst catálogo de espacios de tu equipo con planos personalizados. Según el plano, tu proyecto se crea con recursos específicos. Para obtener más información, consulte [Crear un proyecto con un plano](#) y [Referencia del plano del proyecto](#).

Tras crear un proyecto, puedes aplicar planos adicionales a tu CodeCatalyst proyecto desde el CodeCatalyst catálogo o al catálogo de tu espacio con planos personalizados. Los planos representan componentes arquitectónicos, por lo que se pueden usar varios planos juntos en el proyecto para incorporar las mejores prácticas de su equipo. Esto también le permite asegurarse de que su proyecto esté actualizado con los últimos cambios en los componentes en evolución. Para obtener más información sobre cómo trabajar con planos en su proyecto, consulte [Trabajar con la administración del ciclo de vida como usuario de blueprint](#).

Aplicación y disociación de planos en un proyecto

Puede aplicar varios planos en un proyecto para incorporar componentes funcionales, recursos y gobierno. Sus proyectos pueden admitir varios elementos que se gestionan de forma independiente en planos independientes. La aplicación de planos a un proyecto reduce la necesidad de crear recursos manualmente y hacer que los componentes de software funcionen. Sus proyectos también pueden mantenerse actualizados a medida que evolucionan los requisitos. Si su proyecto ya no requiere los recursos de un plan, puede disociar el plan de su proyecto. Para obtener más información sobre la aplicación de esquemas a su proyecto, consulte. [Trabajar con la administración del ciclo de vida como usuario de blueprint](#)

Temas

- [Aplicar un plano a un proyecto](#)

- [Desasociar un plano de un proyecto](#)

Aplicar un plano a un proyecto

Para aplicar un plano a su proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio y, a continuación, elija el proyecto en el que desee aplicar un plano.
3. En el panel de navegación, elija Blueprints y, a continuación, seleccione Aplicar blueprint.
4. Elija un plano en la pestaña de CodeCatalyst esquemas o un esquema personalizado en la pestaña Planos de espacios y, a continuación, seleccione Siguiente.
5. En Detalles del plano, selecciona una versión del plano en el menú desplegable de la versión objetivo. La última versión se selecciona automáticamente.
6. En Configurar el blueprint, configure los parámetros del blueprint.
7. Revise las diferencias entre la versión actual del blueprint y la versión actualizada. La diferencia que se muestra en una solicitud de extracción muestra los cambios entre la versión actual y la última versión, que es la versión deseada en el momento en que se creó la solicitud de extracción. Si no se muestra ningún cambio, es posible que las versiones sean idénticas o que hayas elegido la misma versión tanto para la versión actual como para la deseada.
8. Cuando estés seguro de que la solicitud de cambios contiene el código y los cambios que deseas revisar, selecciona Aplicar esquema. Después de crear la solicitud de extracción, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a recursos como archivos utilizando el @ signo seguido del nombre del archivo.

Note

El esquema no se aplicará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

Los autores de los planos también pueden aplicar un plano personalizado a los proyectos de espacios específicos en los que no esté disponible el plano para crear nuevos proyectos o aplicarlo

a proyectos existentes. Para obtener más información, consulte [Publicar y aplicar un plano personalizado en espacios y proyectos específicos](#).

Desasociar un plano de un proyecto

Si no desea nuevas actualizaciones de un blueprint, puede desasociar el blueprint de su proyecto. Los recursos y los componentes de software funcionales que se agreguen al proyecto a partir del plan serán los principales del proyecto.

Para desasociar un plano del proyecto

1. [Abra la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En la CodeCatalyst consola, navegue hasta el espacio y, a continuación, elija el proyecto en el que desee desasociar un plano.
3. En el panel de navegación, elija Blueprints (Esquemas).
4. Elija el plano con los recursos que desee desasociar, seleccione el menú desplegable Acciones y, a continuación, elija Desasociar el esquema.
5. Introduzca para confirmar la disociación `confirm`.
6. Seleccione Confirmar.

Actualización de un plano en un proyecto

Si has creado un proyecto con un blueprint o has aplicado un blueprint a un proyecto existente, se te notificará de las nuevas versiones del blueprint. Antes de actualizar la versión del plan mediante una solicitud de extracción aprobada, puedes ver los cambios en el código y los entornos afectados. La gestión del ciclo de vida te permite actualizar uno o más planos aplicados a tu proyecto, de forma que cada plano se pueda actualizar sin que ello afecte a otras áreas del proyecto. También puede anular las actualizaciones de los planos. Para obtener más información, consulte [Trabajar con la administración del ciclo de vida como usuario de blueprint](#).

Para actualizar un blueprint a la versión más reciente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee actualizar la versión de un blueprint.
3. En el panel de control del espacio, elige el proyecto con el plano que quieres actualizar.

4. En el panel de navegación, selecciona Planos y, a continuación, selecciona el botón de opción correspondiente al plano que deseas actualizar.
5. Selecciona el menú desplegable Acciones y, a continuación, selecciona Actualizar versión.
6. En el menú desplegable de la versión de destino, elige la versión que deseas para la actualización. La última versión se selecciona automáticamente.
7. En Configurar el blueprint, configure los parámetros del blueprint.
8. Revise las diferencias entre la versión actual del blueprint y la versión actualizada. La diferencia que se muestra en una solicitud de extracción son los cambios entre la versión actual y la última versión, que es la versión deseada en el momento de crear la solicitud de extracción. Si no se muestra ningún cambio, es posible que las versiones sean idénticas o que hayas elegido la misma versión tanto para la versión actual como para la deseada.
9. Cuando estés seguro de que la solicitud de cambios contiene el código y los cambios que deseas revisar, selecciona Aplicar actualización. Después de crear la solicitud de extracción, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de los archivos y a la solicitud de extracción en general. Puedes añadir enlaces a recursos como archivos utilizando el @ signo seguido del nombre del archivo.

Note

El blueprint no se actualizará hasta que se apruebe y fusione la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

Note

Si tienes solicitudes de extracción existentes abiertas para actualizar una versión del blueprint, cierra las solicitudes de extracción anteriores antes de crear una nueva. Si eliges Actualizar versión, accederás a la lista de solicitudes de extracción pendientes del blueprint. También puedes ver las solicitudes de extracción pendientes en la pestaña Planos de la configuración del proyecto y en la página de resumen del proyecto. Para obtener más información, consulte [Visualización de solicitudes de cambios](#).

Edición de la descripción de un plano de un proyecto

Puedes editar la descripción de un plano que usaste para crear un proyecto o que aplicaste después de crearlo. Un plano se puede utilizar más de una vez en un proyecto. Para diferenciar el propósito de los esquemas en su proyecto, puede usar descripciones para esos esquemas. Las descripciones también se pueden usar para identificar los componentes que se están aplicando a partir de un esquema específico.

Para editar la descripción de un plano en su proyecto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta su espacio y, a continuación, elija el proyecto con la configuración del plano que desee actualizar.
3. En el panel de navegación, elija Blueprints (Esquemas).
4. Elige el plano con la descripción que quieres actualizar, selecciona el menú desplegable Acciones y, a continuación, selecciona Editar descripción.
5. En el campo de entrada del texto de la descripción del plano, introduce una descripción para identificar el plano de tu proyecto.
6. Elija Guardar.

Trabajar con la administración del ciclo de vida como usuario de blueprint

La gestión del ciclo de vida es la capacidad de regenerar una base de código a partir de opciones o versiones actualizadas de un plan. Esto permite al autor de un plan gestionar de forma centralizada el ciclo de vida de desarrollo de software de cada proyecto que contenga un plan concreto. Por ejemplo, introducir una corrección de seguridad en el esquema de una aplicación web permitirá que todos los proyectos que contengan o se hayan creado a partir del esquema de la aplicación web recojan esa corrección automáticamente. Este mismo marco de administración también le permite, como usuario de un blueprint, cambiar las opciones del blueprint una vez seleccionadas.

Temas

- [Utilizar la gestión del ciclo de vida en los proyectos existentes](#)
- [Utilizar la gestión del ciclo de vida en varios planos de un proyecto](#)
- [Trabajar con conflictos en el ciclo de vida de las solicitudes de cambios](#)

- [Optar por no participar en los cambios en la administración del ciclo de vida](#)
- [Anular la gestión del ciclo de vida de un plan en un proyecto](#)

Utilizar la gestión del ciclo de vida en los proyectos existentes

Puede utilizar la gestión del ciclo de vida para los proyectos creados a partir de planos o en proyectos existentes que no estén asociados a ningún plan. Por ejemplo, puede añadir un plan de prácticas de seguridad estándar a una aplicación five-year-old Java que nunca se haya creado a partir de un plan. El esquema genera un flujo de trabajo de escaneo de seguridad y otro código relacionado. Esa parte del código base de la aplicación Java ahora se actualizará automáticamente con las mejores prácticas de su equipo cada vez que se realicen cambios en el plan.

Utilizar la gestión del ciclo de vida en varios planos de un proyecto

Como los planos representan componentes arquitectónicos, a menudo se pueden usar varios planos juntos en el mismo proyecto. Por ejemplo, un proyecto podría estar compuesto por un plan de API web central creado por un ingeniero de plataformas de la empresa, junto con un plan de verificación de versiones creado por el equipo de seguridad de la aplicación. Cada uno de esos esquemas se puede actualizar de forma independiente y recordará las resoluciones de fusión que se les hayan aplicado en el pasado.

Note

Como componentes arquitectónicos arbitrarios, no todos los planos tienen sentido juntos ni funcionarán juntos de forma lógica, aunque sigan intentando fusionarse entre sí.

Trabajar con conflictos en el ciclo de vida de las solicitudes de cambios

En ocasiones, las solicitudes de extracción durante el ciclo de vida pueden generar conflictos de fusión. Estos se pueden resolver manualmente. Las resoluciones se recuerdan en las actualizaciones posteriores del plan.

Optar por no participar en los cambios en la administración del ciclo de vida

Los usuarios pueden eliminar un blueprint de un proyecto para disociar todas las referencias al blueprint y optar por no recibir actualizaciones del ciclo de vida. Por motivos de seguridad, esto no

elimina ni afecta a ninguno de los recursos o códigos del proyecto, incluido lo que se agregó desde el plan. Para obtener más información, consulte [Desasociar un plano de un proyecto](#).

Anular la gestión del ciclo de vida de un plan en un proyecto

Si desea anular las actualizaciones de un blueprint en archivos específicos de su proyecto, puede incluir un archivo de propiedad en su repositorio. GitLabLa especificación [Code Owners](#) es la guía recomendada. El plano siempre respeta el archivo de propietarios del código por encima de todo lo demás y puede generar uno de muestra como el siguiente:

```
new BlueprintOwnershipFile(sourceRepo, {
  resynthesis: {
    strategies: [
      {
        identifier: 'dont-override-sample-code',
        description: 'This strategy is applied accross all sample code. The
blueprint will create sample code, but skip attempting to update it.',
        strategy: MergeStrategies.neverUpdate,
        globs: [
          '**/src/**',
          '**/css/**',
        ],
      },
    ],
  },
});
```

Esto genera un `.ownership-file` con el siguiente contenido:

```
[dont-override-sample-code] @amazon-codecatalyst/blueprints.import-from-git
# This strategy is applied accross all sample code. The blueprint will create sample
code, but skip attempting to update it.
# Internal merge strategy: neverUpdate
**/src/**
**/css/**
```

Referencia del plano del proyecto

Al crear un proyecto con un plano, CodeCatalyst crea un proyecto completo con un repositorio fuente, ejemplos de código fuente, flujos de trabajo de CI/CD, informes de creación y prueba y herramientas integradas de seguimiento de problemas. El plan de un proyecto utiliza el código

para aprovisionar la infraestructura de la nube, los recursos y los artefactos fuente de muestra para distintos tipos de aplicaciones y marcos.

Para obtener más información, consulte [Crear un proyecto en Amazon CodeCatalyst](#). Debe ser el administrador de Space para crear un proyecto.

Temas

- [Planos disponibles](#)
- [Búsqueda de información sobre el plano del proyecto](#)

Planos disponibles

Nombre del plano	Descripción del plano
API web de ASP.NET Core	Este blueprint crea una aplicación de API web ASP.NET Core de .NET 6. El blueprint utiliza la herramienta de AWS implementación para .NET y ofrece una opción para configurar Amazon Elastic Container Service o AWS Elastic Beanstalk como destino de implementación. AWS App Runner
AWS Glue ETL	Este modelo crea un ejemplo de implementación de referencia de carga de extracción y transformación (ETL) con AWS CDK, AWS Glue, AWS Lambda y Amazon Athena para convertir valores separados por comas (CSV) en Apache Parquet.
DevOps canalización de despliegue	Este plan crea una canalización de despliegue e mediante la arquitectura de referencia de la canalización de AWS despliegue que despliega una aplicación de referencia AWS en varias etapas.
API de Java con AWS Fargate	Este plano crea un proyecto de servicio web en contenedores. El proyecto utiliza la CLI

Nombre del plano	Descripción del plano
	<p>de AWS Copilot para crear e implementar un servicio web Spring Boot Java en contenedores respaldado por Amazon DynamoDB en Amazon ECS. El proyecto implementa una aplicación en contenedores en un clúster de Amazon ECS en AWS Fargate un entorno informático sin servidor. La aplicación almacena los datos en una tabla de DynamoDB. Una vez que el flujo de trabajo se ejecute correctamente, el servicio web de muestra estará disponible públicamente a través de Application Load Balancer.</p>
Aplicación web moderna de tres niveles	Este plan genera código en Python para la capa de aplicación y el marco de front-end de Vue a fin de crear e implementar una aplicación web moderna de 3 niveles bien diseñada.
Aplicación sin servidor.NET	Este blueprint crea AWS Lambda funciones mediante las herramientas Lambda de la CLI de .NET. El esquema proporciona opciones para las AWS Lambda funciones, incluida la opción de elegir entre C# o F#.

Nombre del plano	Descripción del plano
API Node.js con AWS Fargate	Este plano crea un proyecto de servicio web en contenedores. El proyecto utiliza la CLI de AWS Copilot para crear e implementar un servicio web Express/Node.js en contenedores en Amazon Elastic Container Service. El proyecto implementa una aplicación en contenedores en un clúster de Amazon ECS en AWS Fargate un entorno informático sin servidor. Una vez que el flujo de trabajo se ejecute correctamente, el servicio web de muestra estará disponible públicamente a través de Application Load Balancer.
Modelo de aplicaciones sin servidor (SAM)	Este plan crea un proyecto que utiliza un modelo de aplicaciones sin servidor (SAM) para crear e implementar una API. Puede elegir SDK for Java o SDK for Python como lenguaje de programación. TypeScript
Gestor de imágenes sin servidor	Este plano crea una aplicación para el procesamiento de imágenes a alta velocidad sin reducir la calidad de la imagen.
Microservicio RESTful sin servidor	Este plan crea una API REST que usa AWS Lambda y Amazon API Gateway con una referencia de servicio de tareas pendientes. Puede elegir SDK for Java o SDK for Python como lenguaje de programación. TypeScript
Aplicación de una sola página	Este plan crea una aplicación de una sola página (SPA) que utiliza los marcos React, Vue y Angular. Para el alojamiento, elija entre AWS Amplify Hosting o Amazon CloudFront Amazon S3.

Nombre del plano	Descripción del plano
Sitio web estático	Este plan crea un sitio web estático utilizando o los generadores de sitios estáticos Hugo o Jekyll . Los generadores de sitios estáticos utilizan archivos de entrada de texto (como Markdown) para generar páginas web estáticas. Son ideales para contenido informativo que rara vez cambia, como páginas de productos, documentación y blogs. El blueprint lo utiliza AWS CDK para implementar páginas web estáticas en Amazon S3 + AWS Amplify o en Amazon S3 + CloudFront.
¿Qué hacer? Aplicación web	Este plan crea una aplicación web sin servidor de tareas pendientes con componentes frontend y backend. Puede elegir SDK for Java o SDK for Python como lenguaje de programación. TypeScript
Servicio video-on-demand web V	Este plan crea un video-on-demand servicio que brinda la capacidad de recibir, transcodificar y entregar contenido. El plano utiliza AWS Lambda Amazon S3 Amazon CloudWatch y AWS Elemental MediaConvert.

Nombre del plano	Descripción del plano
Suscríbase a un plan externo	Este esquema crea un flujo de trabajo para cada paquete importado. Estos flujos de trabajo se ejecutan una vez al día para comprobar si NPM contiene nuevas versiones de los paquetes. Si existe una versión nueva, el flujo de trabajo intentará añadirla a tu CodeCatalyst espacio como un plano personalizado. La acción fallará si no se encuentra un paquete o no es un plano. El paquete de destino debe estar en NPM y debe ser un plano. El espacio debe estar suscrito en un nivel que admita planos personalizados.
Chatbot Bedrock GenAI	Este plan crea un chatbot de IA generativo con Amazon Bedrock y Claude de Anthropic. Con este plan, puede crear e implementar su propio entorno de LLM seguro y protegido mediante el inicio de sesión, que se puede personalizar en función de sus datos. Para obtener más información, consulte la documentación del Chatbot Bedrock GenAI .
Planos del kit de desarrollo de proyectos de AWS (AWS PDK)	Estos planos de PDK se pueden componer juntos para crear una aplicación compuesta por un sitio web de React, la API de Smithy y la infraestructura de CDK de soporte para implementarla en AWS. El PDK de AWS proporciona los componentes básicos de los patrones comunes, junto con herramientas de desarrollo para gestionar y crear sus proyectos . Para obtener más información, consulte el repositorio de GitHub código fuente de AWS PDK y Tutorial: Creación de una aplicación completa con planos de PDK componibles .

Búsqueda de información sobre el plano del proyecto

Hay varios planos de proyectos disponibles en CodeCatalyst. Para cada plano, hay un resumen adjunto y un archivo README. El resumen describe los recursos que instala el blueprint, mientras que el archivo README explica el blueprint en detalle y proporciona instrucciones sobre cómo usarlo.

Trabajar con planos personalizados en CodeCatalyst

Puedes estandarizar el desarrollo y las mejores prácticas para los proyectos de tu CodeCatalyst espacio con planos personalizados. Los planos personalizados se pueden usar para definir varios aspectos de un CodeCatalyst proyecto, como las definiciones del flujo de trabajo y el código de la aplicación. Una vez que se utiliza un esquema personalizado para crear un proyecto nuevo o se aplica a proyectos existentes, cualquier cambio realizado en el esquema estará disponible para esos proyectos como actualizaciones de tipo pull request. Como autor de planos, puedes ver los detalles de los proyectos que utilizan tus planos en todo tu espacio, de forma que puedas ver cómo se aplican los estándares en todos los proyectos. La gestión del ciclo de vida de un plano te permite gestionar de forma centralizada el ciclo de desarrollo de software de cada proyecto, lo que te permite asegurarte de que los proyectos de tu espacio sigan siguiendo las mejores prácticas con los últimos cambios o correcciones. Para obtener más información, consulte [Trabajar con la administración del ciclo de vida como autor de un plan](#).

Los planos personalizados permiten actualizar las versiones de los planos con respecto al proyecto anterior mediante la resíntesis. La resíntesis es el proceso de volver a ejecutar la síntesis de los planos con versiones actualizadas o la posibilidad de incorporar correcciones y cambios en los proyectos existentes. Para obtener más información, consulte [Conceptos de planos personalizados](#).

[Para ver el SDK de planos y los ejemplos de planos, consulta el repositorio de código abierto. GitHub](#)

Temas

- [Conceptos de planos personalizados](#)
- [Cómo empezar con los planos personalizados](#)
- [Tutorial: Creación y actualización de una aplicación React](#)
- [Trabajar con la administración del ciclo de vida como autor de un plan](#)
- [Desarrollo de un plan personalizado](#)
- [Publicar un blueprint personalizado](#)
- [Visualización de los detalles, las versiones y los proyectos de un blueprint personalizado](#)

- [Añadir y eliminar un plano personalizado en un espacio](#)
- [Administrar los permisos de publicación de un blueprint personalizado](#)
- [Administración de versiones para un blueprint personalizado](#)
- [Eliminar un blueprint o una versión personalizados publicados](#)
- [Trabajar con dependencias y herramientas](#)
- [Contribuye](#)

Conceptos de planos personalizados

Estos son algunos conceptos y términos que debe conocer al trabajar con planos personalizados. CodeCatalyst

Temas

- [Proyecto de anteproyecto](#)
- [Planos de espacio](#)
- [Catálogo de planos de espacios](#)
- [Síntesis](#)
- [Resíntesis](#)
- [Opciones parciales](#)
- [Projen](#)

Proyecto de anteproyecto

Un proyecto de anteproyecto te da la posibilidad de desarrollar y publicar planos para tu espacio. Durante el proceso de creación del proyecto, se crea un repositorio de origen y el nombre del repositorio es el que se elige al introducir los detalles de los recursos del proyecto. Durante el proceso de creación del blueprint, si decide generar una versión del flujo de trabajo, se crea un flujo de trabajo de publicación en su blueprint con el blueprint Builder. El flujo de trabajo publica automáticamente su última versión.

Planos de espacio

Puedes ver y gestionar todos los planos desde la tabla de planos de espacios cuando accedas a la sección Planos de tu espacio. Una vez publicados los planos en tu espacio, estarán disponibles como planos de espacio para añadirlos y eliminarlos del catálogo de planos de tu espacio. También

puedes gestionar los permisos de publicación y eliminar los planos desde la sección Planos de tu espacio. Para obtener más información, consulte [Visualización de los detalles, las versiones y los proyectos de un blueprint personalizado](#).

Catálogo de planos de espacios

Puedes ver todos los planos personalizados añadidos desde el catálogo de planos de un espacio. Aquí es donde un miembro del espacio puede elegir su plano personalizado para crear un nuevo proyecto. Este catálogo es diferente del CodeCatalyst catálogo, que ya tiene planos disponibles para todos los miembros del espacio. Para obtener más información, consulte [Referencia del plano del proyecto](#).

Síntesis

La síntesis es el proceso de generar un paquete de CodeCatalyst proyectos que representa el código fuente, la configuración y los recursos de un proyecto. Luego, las operaciones de la API de CodeCatalyst despliegue utilizan el paquete para implementarlo en un proyecto. El proceso se puede ejecutar de forma local mientras se desarrolla un plan personalizado para emular la creación del proyecto sin tener que crear un proyecto en él. CodeCatalyst Se pueden usar los siguientes comandos para realizar la síntesis:

```
yarn blueprint:synth          # fast mode
yarn blueprint:synth --cache  # wizard emulation mode
```

El plano comienza por sí solo llamando a la `blueprint.ts` clase principal con esa opción `fusionadefaults.json`. Se genera un nuevo paquete de proyectos en la `synth/synth.[options-name]/proposed-bundle/` carpeta. El resultado incluye el paquete de proyectos que genera un esquema personalizado, según las opciones que haya establecido, incluidas las [opciones parciales](#) que haya configurado.

Resíntesis

La resíntesis es el proceso de regenerar un plano con diferentes opciones de planos o versiones de planos de proyectos existentes. Como autor de un esquema, puede definir estrategias de combinación personalizadas en el código del esquema personalizado. También puede definir los límites de propiedad `.ownership-file` para especificar en qué partes del código base se permite actualizar un esquema. Si bien el esquema personalizado puede proponer actualizaciones `.ownership-file`, los desarrolladores de proyectos que lo utilicen pueden determinar los límites de propiedad de sus proyectos. Puede ejecutar la resíntesis de forma local y

probarla y actualizarla antes de publicar su esquema personalizado. Utilice los siguientes comandos para realizar la resíntesis:

```
yarn blueprint:resynth          # fast mode
yarn blueprint:resynth --cache  # wizard emulation mode
```

El plano comienza por sí solo llamando a la `blueprint.ts` clase principal con esa opción fusionada. `defaults.json` Se genera un nuevo paquete de proyectos en la `synth/resynth.[options-name]` carpeta. El resultado incluye el paquete de proyectos que genera un esquema personalizado, según las opciones que haya establecido, incluidas las [opciones parciales](#) que haya configurado.

Tras los procesos de síntesis y resíntesis, se crean los siguientes contenidos:

- `paquete propuesto`: el resultado de la síntesis cuando se ejecuta con nuevas opciones para la versión del plan objetivo.
- `existing-bundle`: un simulacro de tu proyecto actual. Si no hay nada en esta carpeta, se genera con el mismo resultado que él. `proposed-bundle`
- `ancestor-bundle`: un simulacro de lo que generaría tu blueprint si lo ejecutaras con una versión anterior, con opciones anteriores o con una combinación de ellas. Si no hay nada en esta carpeta, se genera con el mismo resultado que la. `proposed-bundle`
- `resolved-bundle`: el paquete siempre se regenera y, por defecto, se combina a tres bandas entre, y. `proposed-bundle` `existing-bundle` `ancestor-bundle` Este paquete proporciona una emulación de lo que generaría una resíntesis de forma local.

Para obtener más información sobre los paquetes de salida de blueprint, consulte. [Generar archivos con resíntesis](#)

Opciones parciales

Puede añadir variantes de opciones para no tener `src/wizard-configuration/` que enumerar toda la `Options` interfaz y las opciones se combinan en la parte superior del `defaults.json` archivo. Esto te permite adaptar los casos de prueba a distintas opciones.

Ejemplo:

`Options`interfaz:

```
{
```

```
language: "Python" | "Java" | "Typescript",
repositoryName: string
...
}
```

Archivo `defaults.json`:

```
{
  language: "Python",
  repositoryName: "Myrepo"
  ...
}
```

Pruebas de configuración adicionales:

- `#wizard-config-typescript-test.json`

```
{
  language: "Typescript",
}
```

- `#wizard-config-java-test.json`

```
{
  language: "Java",
}
```

Projen

Projen es una herramienta de código abierto que utilizan los planos personalizados para mantenerse actualizados y coherentes. Los blueprints vienen como paquetes de Projen porque este marco te permite crear, agrupar y publicar proyectos, y puedes usar la interfaz para gestionar las configuraciones y los ajustes de un proyecto.

Puedes usar Projen para actualizar los planos a escala, incluso después de haberlos creado. La herramienta Projen es la tecnología subyacente a la síntesis de planos que genera un paquete de proyectos. Projen es el propietario de la configuración de un proyecto y no debería afectarte a ti como autor de los planos. Puede ejecutar `yarn projen` para regenerar la configuración del proyecto después de añadir las dependencias, o bien puede cambiar las opciones del archivo `projenrc.ts`. Projen también es la herramienta de generación subyacente de planos personalizados para sintetizar un proyecto. [Para obtener más información, consulte la página](#)

[del proyecto. GitHub](#) Para obtener más información sobre cómo trabajar con Projen, consulte la [documentación de Projen](#) y [Cómo simplificar la configuración de un proyecto](#) con Projen.

Cómo empezar con los planos personalizados

Durante el proceso de creación de un plano, puedes configurarlo y generar una vista previa de los recursos del proyecto. Cada plano personalizado lo gestiona un CodeCatalyst proyecto, que contiene un flujo de trabajo de forma predeterminada para publicarlo en el catálogo de planos del espacio.

Temas

- [Requisitos previos](#)
- [Paso 1: Crea un plano personalizado en CodeCatalyst](#)
- [Paso 2: Desarrolle un esquema personalizado con componentes](#)
- [Paso 3: Obtenga una vista previa de un plano personalizado](#)
- [\(Opcional\) Paso 4: Publicar una versión preliminar del blueprint personalizada](#)

Requisitos previos

Antes de crear un plano personalizado, tenga en cuenta los siguientes requisitos:

- Tu CodeCatalyst espacio debe ser del nivel Enterprise. Para obtener más información, consulta [Gestión de la facturación](#) en la Guía del CodeCatalyst administrador de Amazon.
- Debe tener el rol de administrador del espacio o de usuario avanzado para crear planos personalizados. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Paso 1: Crea un plano personalizado en CodeCatalyst

Cuando creas un plano personalizado a partir de la configuración de tu espacio, se crea un repositorio para ti. El repositorio incluye todos los recursos necesarios que debes tener para desarrollar tu plano antes de publicarlo en el catálogo de planos del espacio.

Para crear un plano personalizado

- a. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
- b. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un plano personalizado.

- c. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
- d. Selecciona Crear plano.
- e. En Asigne un nombre a su plano, introduzca el nombre que desee asignar al proyecto y los nombres de los recursos asociados. El nombre debe ser único dentro de su espacio.
- f. En Detalles del plano, haga lo siguiente:
 - i. En el campo de introducción de texto con el nombre para mostrar del plano, introduce un nombre que aparecerá en el catálogo de planos de tu espacio.
 - ii. En el campo de entrada de texto Descripción, introduce una descripción para tu plano personalizado.
 - iii. En el campo de entrada de texto del nombre del autor, introduzca un nombre de autor para el esquema personalizado.
 - iv. (Opcional) Seleccione la configuración avanzada.
 - A. Seleccione + Añadir para añadir las etiquetas que se añadirán al package .json archivo.
 - B. Seleccione el menú desplegable Licencia y, a continuación, elija una licencia para su plano personalizado.
 - C. En el campo de entrada de texto del nombre del paquete de blueprint, introduzca un nombre para identificar su paquete de blueprint.
 - D. De forma predeterminada, el flujo de trabajo de publicación se genera mediante un esquema de publicación del proyecto denominado Blueprint Builder. El flujo de trabajo publica la última versión del blueprint en tu espacio cuando insertas cambios, ya que el flujo de trabajo de publicación habilita los permisos de publicación. Para desactivar la generación del flujo de trabajo, desmarca la casilla Publicar flujo de trabajo.
- g. (Opcional) Un proyecto de plano viene con un código predefinido para facilitar la publicación del plano en el catálogo de espacios. Para ver los archivos de definición con actualizaciones basadas en las selecciones de parámetros del proyecto que haya realizado, seleccione Ver código o Ver flujo de trabajo en la vista previa de Generar plano.
- h. Seleccione Crear plano.

Si no ha desactivado la generación del flujo de trabajo para su esquema personalizado, el flujo de trabajo comienza a ejecutarse automáticamente cuando se crea el esquema. Cuando se complete la ejecución del flujo de trabajo, tu plano personalizado estará disponible para añadirlo al catálogo de planos de tu espacio de forma predeterminada. Puedes desactivar los permisos de publicación si no quieres que la última versión del plano se publique automáticamente en tu espacio. Para obtener más información, consulte [Aplicación y disociación de planos en un proyecto](#).

Como el flujo de trabajo de publicación denominado `blueprint-release` se crea mediante un esquema, el esquema se puede encontrar como un esquema aplicado en el proyecto. Para obtener más información, consulte [Aplicación y disociación de planos en un proyecto](#) y [Trabajar con flujos de trabajo](#).

Paso 2: Desarrolle un esquema personalizado con componentes

Al crear un esquema personalizado, se genera un asistente de planos, que se puede modificar con componentes al desarrollar el esquema personalizado. Puede actualizar los `src/defaults.json` archivos `src/blueprints.js` y para modificar el asistente.

Important

Si desea utilizar paquetes de planos de fuentes externas, tenga en cuenta los riesgos que pueden conllevar dichos paquetes. Eres responsable de los planos personalizados que añadas a tu espacio y del código que generen.

Crea un entorno de desarrollo en tu CodeCatalyst proyecto con un entorno de desarrollo integrado (IDE) compatible antes de configurar el código del blueprint. Se necesita un entorno de desarrollo para trabajar con las herramientas y los paquetes necesarios.

Para crear un entorno de desarrollo

1. En el panel de navegación, realice una de las siguientes acciones:
 - a. Seleccione Descripción general y, a continuación, vaya a la sección Mis entornos de desarrollo.
 - b. Selecciona Código y, a continuación, selecciona Entornos de desarrollo.
 - c. Elija Código, repositorios de código fuente y elija el repositorio que creó al crear su blueprint.

2. Seleccione Crear entorno de desarrollo.
3. Elija un IDE compatible en el menú desplegable. Consulte Entornos de [desarrollo integrados compatibles para entornos de desarrollo](#) para obtener más información.
4. Seleccione Trabajar en una rama existente y, en el menú desplegable de la rama existente, seleccione la rama de funciones que has creado.
5. (Opcional) En el campo de entrada de texto Alias: opcional, introduce un alias para identificar el entorno de desarrollo.
6. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo muestra Inicio y la columna de estado muestra En ejecución cuando se ha creado el entorno de desarrollo.

Para obtener más información, consulte [Entornos de desarrollo en CodeCatalyst](#).

Para desarrollar su plan personalizado

1. En una terminal que funcione, utilice el siguiente yarn comando para instalar las dependencias:

```
yarn
```

Las herramientas y los paquetes necesarios están disponibles a través del entorno de CodeCatalyst desarrollo, incluido Yarn. Si está trabajando en un plan personalizado sin un entorno de desarrollo, instale primero Yarn en su sistema. Para obtener más información, consulta la [documentación de instalación de Yarn](#).

2. Desarrolle su plano personalizado para que se configure según sus preferencias. Puede modificar el asistente del blueprint añadiendo componentes. Para obtener más información, consulte [Desarrollo de un plan personalizado](#), [Trabajar con un asistente](#) y [Publicar un blueprint personalizado](#).

Paso 3: Obtenga una vista previa de un plano personalizado

Tras configurar y desarrollar tu plano personalizado, puedes previsualizar y publicar la versión preliminar del plano en tu espacio. Una versión de vista previa te permite comprobar si el plano es lo que deseas antes de usarlo para crear nuevos proyectos o aplicarlo a proyectos existentes.

Para obtener una vista previa de un plano personalizado

1. En una terminal que funcione, utilice el siguiente yarn comando:

```
yarn blueprint:preview
```

2. Navegue hasta el `See this blueprint at:` enlace proporcionado para obtener una vista previa del plano personalizado.
3. Comprueba que la interfaz de usuario, incluido el texto, se muestre como esperabas en función de tu configuración. Si desea cambiar el esquema personalizado, puede editar el `blueprint.ts` archivo, volver a sintetizar el esquema y, a continuación, volver a publicar una versión preliminar. Para obtener más información, consulte [Resíntesis](#).

(Opcional) Paso 4: Publicar una versión preliminar del blueprint personalizada

Puedes publicar una versión preliminar de tu plano personalizado en tu espacio si quieres añadirlo al catálogo de planos de tu espacio. Esto le permite ver el plano como usuario antes de añadir la versión sin vista previa al catálogo. La versión preliminar le permite publicar sin ocupar una versión real. Por ejemplo, si trabaja en una `0.0.1` versión, puede publicar y añadir una versión preliminar, de forma que las nuevas actualizaciones de una segunda versión puedan publicarse y añadirse como tal `0.0.2`.

Para publicar una versión preliminar de un blueprint personalizado

Navegue hasta el `Enable version [version number] at:` enlace proporcionado para activar su esquema personalizado. Este enlace se proporciona al ejecutar el `yarn` comando en [Paso 3: Obtenga una vista previa de un plano personalizado](#).

Tras crear, desarrollar, previsualizar y publicar tu plano personalizado, puedes publicar y añadir la versión final del plano al catálogo de planos de tu espacio. Para obtener más información, consulte [Añadir y eliminar un plano personalizado en un espacio](#).

Tutorial: Creación y actualización de una aplicación React

Como autor de planos, puedes desarrollar y añadir planos personalizados al catálogo de planos de tu espacio. Luego, los miembros del espacio pueden usar estos planos para crear nuevos proyectos o aplicarlos a proyectos existentes. Puedes seguir realizando cambios en tus planos para que luego estén disponibles como actualizaciones a través de solicitudes de extracción de información.

En este tutorial se ofrece un recorrido desde la perspectiva del autor del plano y del usuario del mismo. El tutorial muestra cómo crear un plano de aplicación web de una sola página para React.

Luego, el plano se usa para crear un nuevo proyecto. Cuando el blueprint se actualiza con los cambios, el proyecto creado a partir del blueprint incorpora esos cambios mediante una solicitud de extracción.

Temas

- [Requisitos previos](#)
- [Paso 1: Crea un plano personalizado](#)
- [Paso 2: Ver el flujo de trabajo de las versiones](#)
- [Paso 3: Añadir el plano al catálogo](#)
- [Paso 4: Crea un proyecto con blueprint](#)
- [Paso 5: Actualizar el esquema](#)
- [Paso 6: Actualice la versión del catálogo publicada del blueprint a la nueva versión](#)
- [Paso 7: Actualice el proyecto con una nueva versión del plan](#)
- [Paso 8: Ver los cambios en el proyecto](#)

Requisitos previos

Para crear y actualizar un esquema personalizado, debes haber completado las tareas de la siguiente manera [Configuración CodeCatalyst](#):

- Tenga un ID de AWS constructor para iniciar CodeCatalyst sesión.
- Pertenece a un espacio y tienes asignado el rol de administrador del espacio o usuario avanzado en ese espacio. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#), [Gestión de los usuarios del espacio](#) y [Función de administrador de espacios](#).

Paso 1: Crea un plano personalizado

Al crear un plan personalizado, se crea un CodeCatalyst proyecto que contiene el código fuente del plan y las herramientas y los recursos de desarrollo. Tu proyecto es el lugar donde desarrollarás, probarás y publicarás el plan.

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un plano.
3. Seleccione Configuración para ir a la configuración del espacio.

4. En la pestaña de configuración del espacio, selecciona Planos y selecciona Crear plano.
5. Actualice los campos del asistente de creación de planos con los siguientes valores:
 - En el nombre del plano, introduzca. `react-app-blueprint`
 - En Nombre para mostrar del esquema, introduzca. `react-app-blueprint`
6. Si lo desea, seleccione Ver código para obtener una vista previa del código fuente del blueprint de su blueprint. Del mismo modo, elija Ver flujo de trabajo para obtener una vista previa del flujo de trabajo que se creará en el proyecto que crea y publica el blueprint.
7. Elija Crear esquema.
8. Una vez creado el plano, se le redirigirá al proyecto del plano. Este proyecto contiene el código fuente del plan, junto con las herramientas y los recursos que necesita para desarrollar, probar y publicar el plan. Se generó un flujo de trabajo de publicación que publicó automáticamente tu plan en el espacio.
9. Ahora que el plano y el proyecto de anteproyecto están creados, el siguiente paso es configurarlos actualizando el código fuente. Puedes usar Dev Environments para abrir y editar tu repositorio de código fuente directamente en tu navegador.

En el panel de navegación, selecciona Código y, a continuación, selecciona Entornos de desarrollo.

10. Elija Crear entorno de desarrollo y, a continuación, elija AWS Cloud9 (en el navegador).
11. Mantenga la configuración predeterminada y elija Crear.
12. En la AWS Cloud9 terminal, navegue hasta el directorio de su proyecto de blueprint ejecutando el siguiente comando:

```
cd react-app-blueprint
```

13. Al crear un plano, se crea y rellena automáticamente una `static-assets` carpeta. En este tutorial, eliminarás la carpeta predeterminada y generarás una nueva para el blueprint de una aplicación de React.

Elimine la carpeta `static-assets` ejecutando el siguiente comando:

```
rm -r static-assets
```

AWS Cloud9 se basa en una plataforma basada en Linux. Si utilizas un sistema operativo Windows, puedes usar el siguiente comando en su lugar:

```
rmdir /s /q static-assets
```

14. Ahora que se ha eliminado la carpeta predeterminada, cree una `static-assets` carpeta para un blueprint de react-app ejecutando el siguiente comando:

```
npx create-react-app static-assets
```

Si se te solicita, ingresa `y` para continuar.

Se creó una nueva aplicación de reacción en la `static-assets` carpeta con los paquetes necesarios. Los cambios deben enviarse a su repositorio de CodeCatalyst código fuente remoto.

15. Asegúrese de tener los cambios más recientes y, a continuación, confirme e inserte los cambios en el repositorio CodeCatalyst fuente del blueprint ejecutando los siguientes comandos:

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Cuando se envía un cambio al repositorio de origen del blueprint, el flujo de trabajo de publicación se inicia automáticamente. Este flujo de trabajo incrementa la versión del blueprint, crea el blueprint y lo publica en tu espacio. En el siguiente paso, navegarás hasta la ejecución del flujo de trabajo de la versión para ver su rendimiento.

Paso 2: Ver el flujo de trabajo de las versiones

1. En la CodeCatalyst consola, en el panel de navegación, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
2. Elija el flujo de trabajo del blueprint-release.
3. Puede ver que el flujo de trabajo incluye acciones para crear y publicar el blueprint.

4. En Última ejecución, elija el enlace de ejecución del flujo de trabajo para ver la ejecución a partir del cambio de código que realizó.
5. Una vez finalizada la ejecución, se publica la nueva versión del plan. Las versiones de planos publicadas se pueden ver en la configuración del espacio, pero no se pueden usar en proyectos hasta que se agreguen al catálogo de planos del espacio. En el siguiente paso, añadirá el plano al catálogo.

Paso 3: Añadir el plano al catálogo

Al añadir un plano al catálogo de planos del espacio, el plano estará disponible para su uso en todos los proyectos de un espacio. Los miembros del espacio pueden usar el plano para crear nuevos proyectos o aplicarlo a proyectos existentes.

1. En la CodeCatalyst consola, vuelve al espacio.
2. Selecciona Configuración y, a continuación, selecciona Planos.
3. Elija y react-app-blueprint, a continuación, elija Añadir al catálogo.
4. Seleccione Guardar.

Paso 4: Crea un proyecto con blueprint

Ahora que el plano se ha añadido al catálogo, se puede utilizar en proyectos. En este paso, creará un proyecto con el plano que acaba de crear. En un paso posterior, actualizará este proyecto actualizando y publicando una nueva versión del blueprint.

1. Seleccione la pestaña Proyectos y, a continuación, elija Crear proyecto.
2. Seleccione Planos de espacio y, a continuación, elija react-app-blueprint.

Note

Una vez elegido el plano, podrás ver el contenido del archivo del plano. README .md

3. Elija Siguiente.

4.

Note

El contenido de este asistente de creación de proyectos se puede configurar en el plano.

Introduzca el nombre del proyecto como usuario del blueprint. En este tutorial, escriba `react-app-project`. Para obtener más información, consulte [Desarrollo de un plan personalizado](#).

A continuación, actualizará el esquema y agregará la nueva versión al catálogo, que utilizará para actualizar este proyecto.

Paso 5: Actualizar el esquema

Una vez que se haya utilizado un plano para crear un proyecto nuevo o se haya aplicado a proyectos existentes, podrá seguir realizando actualizaciones como autor del modelo. En este paso, realizará cambios en el esquema y publicará automáticamente una nueva versión en el espacio. A continuación, la nueva versión se puede añadir como versión del catálogo.

1. Navegue hasta el `react-app-blueprint` proyecto creado en [Tutorial: Creación y actualización de una aplicación React](#).
2. Abra el entorno de desarrollo creado en [Tutorial: Creación y actualización de una aplicación React](#).
 - a. En el panel de navegación, elija Código y, a continuación, elija Entornos de desarrollo.
 - b. En la tabla, busca el entorno de desarrollo y, a continuación, selecciona Abrir en AWS Cloud9 (en el navegador).
3. Cuando se ejecutó el flujo de trabajo de publicación del blueprint, se incrementó la versión del blueprint al actualizar el archivo `package.json`. Para obtener ese cambio, ejecute el siguiente comando en la terminal: AWS Cloud9

```
git pull
```

4. Navegue hasta la `static-assets` carpeta ejecutando el siguiente comando:

```
cd /projects/react-app-blueprint/static-assets
```

5. Cree un `hello-world.txt` archivo en `static-assets` la carpeta ejecutando el siguiente comando:

```
touch hello-world.txt
```

AWS Cloud9 se basa en una plataforma basada en Linux. Si utilizas un sistema operativo Windows, puedes usar el siguiente comando en su lugar:

```
echo > hello-world.txt
```

6. En el panel de navegación de la izquierda, haz doble clic en el `hello-world.txt` archivo para abrirlo en el editor y añade el siguiente contenido:

```
Hello, world!
```

Guarde el archivo.

7. Asegúrese de tener los cambios más recientes y, a continuación, confirme e inserte los cambios en el repositorio CodeCatalyst fuente del blueprint ejecutando los siguientes comandos:

```
git pull
```

```
git add .
```

```
git commit -m "prettier setup"
```

```
git push
```

Al introducir los cambios, se inició el flujo de trabajo de publicación, que publicará automáticamente la nueva versión del blueprint en el espacio.

Paso 6: Actualice la versión del catálogo publicada del blueprint a la nueva versión

Una vez que se haya utilizado un plano para crear un proyecto nuevo o se haya aplicado a proyectos existentes, podrá seguir actualizándolo como autor del modelo. En este paso, realizará cambios en el esquema y cambiará la versión del catálogo del esquema.

1. En la CodeCatalyst consola, vuelve al espacio.
2. Selecciona Configuración y, a continuación, selecciona Planos.
3. Elija y `react-app-blueprint`, a continuación, elija Administrar la versión del catálogo.
4. Elija la nueva versión y, a continuación, seleccione Guardar.

Paso 7: Actualice el proyecto con una nueva versión del plan

Ya hay una nueva versión disponible en el catálogo de planos del espacio. Como usuario de blueprint, puede actualizar la versión del proyecto creado en [Paso 4: Crea un proyecto con blueprint](#). Esto garantiza que dispone de los cambios y correcciones más recientes necesarios para cumplir con las prácticas recomendadas.

1. En la CodeCatalyst consola, navegue hasta el react-app-projectproyecto creado en [Paso 4: Crea un proyecto con blueprint](#).
2. En el panel de navegación, elija Blueprints (Esquemas).
3. Seleccione Actualizar esquema en el cuadro de información.
4. En el panel de cambios de código de la derecha, puedes ver las actualizaciones `hello-world.txt` y `package.json` las actualizaciones.
5. Seleccione Aplicar actualización.

Al seleccionar Aplicar actualización, se crea una solicitud de cambios en el proyecto con los cambios respecto a la versión actualizada del blueprint. Para realizar las actualizaciones en el proyecto, debes fusionar la solicitud de extracción. Para obtener más información, consulte [Revisión de una solicitud de extracción](#) y [Fusión de una solicitud de extracción](#).

1. En la tabla de planos, busca el plano. En la columna Estado, selecciona Solicitud de extracción pendiente y, a continuación, selecciona el enlace a la solicitud de extracción abierta.
2. Revisa la solicitud de extracción y, a continuación, selecciona Fusionar.
3. Elija Fusión rápida para mantener los valores predeterminados y, a continuación, elija Fusionar.

Paso 8: Ver los cambios en el proyecto

Los cambios en el plano ya estarán disponibles en su proyecto posteriormente [Paso 7: Actualice el proyecto con una nueva versión del plan](#). Como usuario del blueprint, puede ver los cambios en el repositorio de origen.

1. En el panel de navegación, elija Repositorios de origen y, a continuación, elija el nombre del repositorio de origen creado cuando se creó el proyecto.
2. En Archivos, puede ver el `hello-world.txt` archivo en [Paso 5: Actualizar el esquema](#) el que se creó.
3. Elija esta opción `hello-world.txt` para ver el contenido del archivo.

La gestión del ciclo de vida proporciona a los autores de los planos la posibilidad de gestionar de forma centralizada el ciclo de vida de desarrollo de software de cada proyecto que contenga un plan concreto. Como se muestra en este tutorial, puede introducir actualizaciones en el plan para luego incorporarlas en los proyectos que lo hayan utilizado para crear un nuevo proyecto o lo hayan aplicado a un proyecto existente. Para obtener más información, consulte [Trabajar con la administración del ciclo de vida como autor de un plan](#).

Trabajar con la administración del ciclo de vida como autor de un plan

La gestión del ciclo de vida te permite mantener sincronizados un gran número de proyectos a partir de una única fuente común de mejores prácticas. Esto amplía la propagación de las correcciones y el mantenimiento de cualquier número de proyectos a lo largo de todo su ciclo de vida de desarrollo de software. La gestión del ciclo de vida optimiza las campañas internas, las correcciones de seguridad, las auditorías, las actualizaciones en tiempo de ejecución, los cambios en las mejores prácticas y otras prácticas de mantenimiento, ya que esos estándares se definen en un solo lugar y se actualizan automáticamente de forma centralizada cuando se publican nuevos estándares.

Cuando se publica una nueva versión del plan, se solicita a todos los proyectos que lo contengan que se actualicen a la versión más reciente. Como autor de un blueprint, también puede ver la versión de un blueprint concreto que contiene cada proyecto para garantizar el cumplimiento. Cuando hay conflictos en un repositorio de origen existente, la gestión del ciclo de vida crea solicitudes de cambios. Para todos los demás recursos, como Dev Environment, todas las actualizaciones de la administración del ciclo de vida crean estrictamente recursos nuevos. Los usuarios son libres de combinar o no esas solicitudes de cambios. Cuando se fusionan las solicitudes de cambios pendientes, se actualiza la versión del plan, incluidas las opciones, utilizada en tu proyecto. Para obtener más información sobre cómo trabajar con la administración del ciclo de vida como usuario de blueprint, consulte [Utilizar la gestión del ciclo de vida en los proyectos existentes](#) y [Utilizar la gestión del ciclo de vida en varios planos de un proyecto](#).

Temas

- [Gestión del ciclo de vida de pruebas](#)
- [Trabajar con estrategias de fusión](#)
- [Trabajar con objetos de contexto](#)

Gestión del ciclo de vida de pruebas

Puede probar localmente la gestión del ciclo de vida de su plan y combinar la resolución de conflictos. En el `synth/` directorio se genera una serie de paquetes que representan las distintas fases de una actualización del ciclo de vida. Para probar la gestión del ciclo de vida, puede ejecutar el siguiente comando `yarn` en su blueprint: `yarn blueprint: resynth` Para obtener más información sobre la resíntesis y los paquetes, consulte y. [Resíntesis](#) [Generar archivos con resíntesis](#)

Trabajar con estrategias de fusión

Temas

- [Generar archivos con resíntesis](#)
- [Uso de estrategias de fusión](#)
- [Especificar archivos para las actualizaciones de la administración del ciclo de vida](#)
- [Escribir estrategias de fusión](#)

Generar archivos con resíntesis

La resíntesis puede combinar el código fuente producido por un plano con el código fuente generado previamente por el mismo modelo, lo que permite propagar los cambios en un plano a los proyectos existentes. Las fusiones se ejecutan desde la función en todos los paquetes de salida del blueprint. `resynth()` La resíntesis primero genera tres paquetes que representan diferentes aspectos del plano y del estado del proyecto. Se puede ejecutar manualmente de forma local con el `yarn blueprint:resynth` comando, que creará los paquetes si aún no existen. Trabajar manualmente con los paquetes le permitirá simular y probar el comportamiento de la resíntesis localmente. De forma predeterminada, los blueprints solo ejecutan la resíntesis en los repositorios en los que se encuentran, `src/*` ya que normalmente solo esa parte del paquete está bajo el control de código fuente.

- `existing-bundle`- Este paquete es una representación del estado del proyecto existente. El cálculo de síntesis lo construye artificialmente para dar al modelo un contexto sobre el contenido del proyecto en el que se va a implementar (si es que hay algo). Si algo ya existe en esta ubicación al ejecutar la resíntesis localmente, se restablecerá y se respetará como un simulacro. De lo contrario, se ajustará al contenido de `ancestor-bundle`
- `ancestor-bundle`- Este es el paquete que representa el resultado del plano si se sintetizó con algunas opciones o versiones anteriores. Si es la primera vez que se agrega este plano a un proyecto, entonces el antepasado no existe, por lo que tiene el mismo contenido que el.

`existing-bundle` A nivel local, si este paquete ya existe en esta ubicación, se respetará como un simulacro.

- `proposed-bundle`- Este es el paquete que se burla del modelo si se ha sintetizado con nuevas opciones o versiones. Es el mismo paquete que generaría la `synth()` función. A nivel local, este paquete siempre se anula.

Cada paquete se crea durante una fase de resíntesis a la que se puede acceder desde la clase de blueprint que aparece a continuación. `this.context.resynthesisPhase`

- `resolved-bundle`- Este es el paquete final, que es una representación de lo que se empaqueta y se implementa en un proyecto. CodeCatalyst Puede ver qué archivos y diferencias se envían a los mecanismos de despliegue. Este es el resultado de la `resynth()` función que resuelve las fusiones entre los otros tres paquetes.

La combinación a tres bandas se aplica tomando la diferencia entre `ancestor-bundle` y `proposed-bundle` y aplicándola a la para generar `existing-bundle`. `resolved-bundle` Todas las estrategias de fusión resuelven los `resolved-bundle` archivos en. La resíntesis resuelve el alcance de estos paquetes con las estrategias de fusión del plan durante el proceso `resynth()` y produce el paquete resuelto a partir del resultado.

Uso de estrategias de fusión

Puede utilizar una estrategia de fusión que ofrece la biblioteca de planos. Estas estrategias proporcionan formas de resolver las salidas de archivos y los conflictos relacionados con los archivos mencionados en la [Generar archivos con resíntesis](#) sección.

- `alwaysUpdate`- Una estrategia que siempre se dirija al archivo propuesto.
- `neverUpdate`- Una estrategia que siempre se dirija al archivo existente.
- `onlyAdd`- Una estrategia que se convierte en el archivo propuesto cuando aún no existe un archivo existente. De lo contrario, se resuelve en el archivo existente.
- `threeWayMerge`- Una estrategia que realiza una fusión a tres bandas entre los archivos existentes, propuestos y ancestros comunes. El archivo resuelto puede contener marcadores de conflicto si los archivos no se pueden combinar de forma limpia. El contenido de los archivos proporcionados debe estar codificado en UTF-8 para que la estrategia produzca resultados significativos. La estrategia intenta detectar si los archivos de entrada son binarios. Si la estrategia detecta un conflicto de fusión en un archivo binario, siempre devuelve el archivo propuesto.

- `preferProposed`- Una estrategia que realiza una fusión a tres bandas entre los archivos existentes, los propuestos y los archivos ancestrales comunes. Esta estrategia resuelve los conflictos seleccionando el lado del archivo propuesto en cada conflicto.
- `preferExisting`- Una estrategia que realiza una fusión a tres bandas entre los archivos existentes, los propuestos y los archivos ancestrales comunes. Esta estrategia resuelve los conflictos seleccionando el lado del archivo existente en cada conflicto.

Para ver el código fuente de las estrategias de fusión, consulta el [GitHub repositorio de código abierto](#).

Especificar archivos para las actualizaciones de la administración del ciclo de vida

Durante la resíntesis, los planos controlan cómo se combinan los cambios en un repositorio de origen existente. Sin embargo, es posible que no desee enviar actualizaciones a todos los archivos del blueprint. Por ejemplo, el código de muestra, como las hojas de estilo CSS, está pensado para ser específico del proyecto. La estrategia de fusión a tres bandas es la opción predeterminada si no especificas otra estrategia. Los blueprints pueden especificar qué archivos son de su propiedad y cuáles no especificando las estrategias de combinación en la propia construcción del repositorio. Los planos pueden actualizar sus estrategias de fusión y las estrategias más recientes se pueden utilizar durante la resíntesis.

```
const sourceRepo = new SourceRepository(this, {
  title: 'my-repo',
});
sourceRepo.setResynthStrategies([
  {
    identifier: 'dont-override-sample-code',
    description: 'This strategy is applied accross all sample code. The blueprint
will create sample code, but skip attempting to update it.',
    strategy: MergeStrategies.neverUpdate,
    globs: [
      '**/src/**',
      '**/css/**',
    ],
  },
]);
```

Se pueden especificar varias estrategias de fusión, y la última estrategia tiene prioridad. Los archivos descubiertos `three-way-merge` tienen un valor predeterminado similar a Git. La `MergeStrategies`

construcción incluye varias estrategias de fusión, pero puedes escribir las tuyas propias. Las estrategias proporcionadas se adhieren al motor de [estrategia de fusión de git](#).

Escribir estrategias de fusión

Además de utilizar una de las estrategias de construcción y fusión que se proporcionan, también puedes escribir tus propias estrategias. Las estrategias deben seguir una interfaz de estrategia estándar. Debe escribir una función de estrategia que tome versiones de un archivo de `existing-bundle`, `proposed-bundle`, `yancestor-bundle`, y las combine en un único archivo resuelto. Por ejemplo:

```
type StrategyFunction = (
  /**
   * file from the ancestor bundle (if it exists)
   */
  commonAncestorFile: ContextFile | undefined,
  /**
   * file from the existing bundle (if it exists)
   */
  existingFile: ContextFile | undefined,
  /**
   * file from the proposed bundle (if it exists)
   */
  proposedFile: ContextFile | undefined,
  options?: {})
  /**
   * Return: file you'd like in the resolved bundle
   * passing undefined will delete the file from the resolved bundle
   */
=> ContextFile | undefined;
```

Si los archivos no existen (no están definidos), entonces la ruta del archivo no existe en ese paquete de ubicaciones en particular.

Ejemplo:

```
strategies: [
  {
    identifier: 'dont-override-sample-code',
    description: 'This strategy is applied across all sample code. The
    blueprint will create sample code, but skip attempting to update it.',
    strategy: (ancestor, existing, proposed) => {
```

```
        const resolvedfile = ...
        ...
        // do something
        ...
        return resolvedfile
    },
    globs: [
        '**/src/**',
        '**/css/**',
    ],
},
],
```

Trabajar con objetos de contexto

Como autor de un plano, puede acceder al contexto del proyecto del plano durante la síntesis para obtener información como los nombres de los espacios y los proyectos, o los archivos existentes en el repositorio de origen de un proyecto. También puede obtener detalles como la fase de resíntesis que está generando el plano. Por ejemplo, puedes acceder al contexto para saber si vas a resintetizar para generar un paquete anterior o un paquete propuesto. Luego, el contexto de código existente se puede usar para transformar el código en el repositorio. Por ejemplo, puedes escribir tu propia estrategia de resíntesis para establecer estándares de código específicos. La estrategia se puede añadir al `blueprint.ts` archivo para los planos pequeños, o se puede crear un archivo independiente para las estrategias.

El siguiente ejemplo muestra cómo encontrar archivos en el contexto de un proyecto, configurar un generador de flujos de trabajo y establecer una estrategia de resíntesis basada en planos para un archivo concreto:

```
const contextFiles = this.context.project.src.findAll({
  fileGlobs: ['**/package.json'],
});

// const workflows = this.context.project.src.findAll({
//   fileGlobs: ['**/.codecatalyst/**/*.yaml'],
// });

const security = new WorkflowBuilder(this, {
  Name: 'security-workflow',
});
new Workflow(this, repo, security.getDefinition());
```

```
repo.setResynthStrategies([
  {
    identifier: 'force-security',
    globs: ['**/.codecatalyst/security-workflow.yaml'],
    strategy: MergeStrategies.alwaysUpdate,
  },
]);

for (const contextFile of contextFiles) {
  const packageObject = JSON.parse(contextFile.buffer.toString());
  new SourceFile(internalRepo, contextFile.path, JSON.stringify({
    ...packageObject,
  }, null, 2));
}
}
```

Desarrollo de un plan personalizado

Antes de publicar un plano personalizado, puedes desarrollarlo para que cumpla con requisitos específicos. Puede desarrollar su esquema personalizado y probarlo creando un proyecto durante la vista previa. Puede desarrollar el esquema personalizado para incluir los componentes del proyecto, como el código fuente específico, las conexiones entre cuentas, los flujos de trabajo, los problemas o cualquier otro componente en el que se pueda crear. CodeCatalyst

Important

Si desea utilizar paquetes de planos de fuentes externas, tenga en cuenta los riesgos que pueden conllevar dichos paquetes. Eres responsable de los planos personalizados que añadas a tu espacio y del código que generen.

Para desarrollar o actualizar un plano personalizado

1. Reanude su entorno de desarrollo. Para obtener más información, consulte [Reanudación de un entorno de desarrollo](#).

Si no tiene un entorno de desarrollo, primero debe crear uno. Para obtener más información, consulte [Creación de un entorno de desarrollo](#).

2. Abre una terminal que funcione en tu entorno de desarrollo.

3. Si optó por un flujo de trabajo de publicación al crear su blueprint, la última versión del blueprint se publicará automáticamente. Extraiga los cambios para asegurarse de que el `package.json` archivo tenga la versión incrementada. Utilice el siguiente comando:

```
git pull
```

4. En el `src/blueprint.ts` archivo, edite las opciones del plano personalizado. El CodeCatalyst asistente interpreta la `Options` interfaz de forma dinámica para generar una interfaz de usuario (UI) de selección. Puede desarrollar su esquema personalizado añadiendo componentes y etiquetas compatibles. Para obtener más información, consulte [Trabajar con un asistente](#), [Componentes del entorno](#), [Componentes de la región](#), [Componentes de repositorio y código fuente](#), [Componentes del flujo de trabajo](#) y [Componentes de entornos de desarrollo](#).

También puede ver el SDK de planos y ejemplos de planos para obtener asistencia adicional a la hora de desarrollar su plan personalizado. [Para obtener más información, consulta el repositorio de código abierto. GitHub](#)

Trabajar con un asistente

La `Options` interfaz del archivo genera automáticamente CodeCatalyst un asistente de selección de planos activado. `blueprint.ts` El asistente de interfaz admite las modificaciones y funciones de un plano `Options` mediante comentarios y etiquetas de estilo [JSDOC](#). Puede utilizar comentarios y etiquetas de estilo JSDOC para realizar tareas. Por ejemplo, puede seleccionar el texto que se muestra encima de una opción, habilitar funciones como la validación de entradas o hacer que una opción sea plegable. El asistente interpreta un árbol sintáctico abstracto (AST) generado a partir del TypeScript tipo de texto de la `Options` interfaz. El asistente se configura automáticamente según el tipo descrito de la mejor manera posible. No se admiten todos los tipos. Otros tipos compatibles incluyen el selector de región y el selector de entorno.

El siguiente es un ejemplo de un asistente que utiliza comentarios y etiquetas del JSDOC con los planos: `Options`

```
export interface Options {  
  /**  
   * What do you want to call your new blueprint?  
   * @validationRegex /^[a-zA-Z0-9_]+$/  
   * @validationMessage Must contain only upper and lowercase letters, numbers and  
   underscores  
   */  
}
```

```
blueprintName: string;

/**
 * Add a description for your new blueprint.
 */
description?: string;

/**
 * Tags for your Blueprint:
 * @collapsed true
 */
tags?: string[];
}
```

El nombre para mostrar de cada opción de la `Options` interfaz aparece de forma predeterminada. `camelCase` El texto sin formato del comentario de estilo JSDOC se muestra como texto encima de la opción en el asistente.

Temas

- [Etiquetas compatibles](#)
- [TypeScript Tipos compatibles](#)
- [Comunicarse con el usuario durante la síntesis](#)

Etiquetas compatibles

Las siguientes etiquetas JSDOC son compatibles con un esquema personalizado `Options` en el asistente de front-end.

`@inlinePolicy. /path/to/policy/file.json`

- **Requiere:** la opción debe ser un tipo. `Role`
- **Uso:** le permite comunicar las políticas integradas que necesita un puesto. Se espera que la `policy.json` ruta esté bajo el código fuente. Use esta etiqueta cuando necesite una política personalizada para un rol.
- **Dependencias,** `blueprint-cli 0.1.12` y más
- **Ejemplo -** `@inlinePolicy ./deployment-policy.json`

```
environment: EnvironmentDefinition{
```

```
awsAccountConnection: AccountConnection{
  /**
   * @inlinePolicy ./path/to/deployment-policy.json
   */
  cdkRole: Role[];
};
};
```

@trustPolicy. /path/to/policy/file.json

- Requiere: la opción debe ser un tipo. Role
- Uso: le permite comunicar las políticas de confianza que necesita un puesto. Se espera que la `policy.json` ruta esté bajo el código fuente. Use esta etiqueta cuando necesite una política personalizada para un rol.
- Dependencias, `blueprint-cli 0.1.12` y más
- Ejemplo - `@trustPolicy ./trust-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @trustPolicy ./path/to/trust-policy.json
     */
    cdkRole: Role[];
  };
};
```

Expresión regex de @validationRegex

- Requiere que la opción sea una cadena.
- Uso: realiza la validación de entrada de la opción mediante la expresión regular dada y se muestra. `@validationMessage`
- Ejemplo: `@validationRegex /^[a-zA-Z0-9_]+$`
- Recomendación: úselo con `@validationMessage`. El mensaje de validación está vacío de forma predeterminada.

Cadena @validationMessage

- Requiere... `@validationRegex` u otros errores para revisar el uso.

- Uso: muestra un mensaje de validación en `@validation*` caso de error.
- Ejemplo `@validationMessage Must contain only upper and lowercase letters, numbers, and underscores.`
- Recomendación: úselo con `@validationMessage`. El mensaje de validación está vacío de forma predeterminada.

@collapsed boolean (opcional)

- Requiere: N/A
- Uso: booleano que permite plegar una subopción. Si la anotación contraída está presente, su valor predeterminado es `true`. Si se establece el valor en, se `@collapsed false` crea una sección plegable que inicialmente está abierta.
- Ejemplo: `@collapsed true`

cadena @displayName

- Requiere: N/A
- Uso: cambia el nombre para mostrar de la opción. Admite formatos distintos de CamelCase para el nombre mostrado.
- Ejemplo - `@displayName Blueprint Name`

cadena @displayName

- Requiere: N/A
- Uso: cambia el nombre para mostrar de la opción. Admite formatos distintos de [CamelCase](#) para el nombre mostrado.
- Ejemplo - `@displayName Blueprint Name`

Número @defaultEntropy

- Requiere que la opción sea una cadena.
- Uso: agrega a la opción una cadena alfanumérica aleatoria de una longitud específica.
- Ejemplo: `@defaultEntropy 5`

@placeholder string (opcional)

- Requiere: N/A
- Uso: cambia el marcador de posición del campo de texto predeterminado.
- Ejemplo - @placeholder type project name here

Número @textArea (opcional)

- Requiere: N/A
- Uso: convierte la entrada de cadena en un componente de área de texto para secciones de texto más grandes. Al añadir un número se define el número de filas. El valor por defecto es de cinco filas.
- Ejemplo - @textArea 10

@hidden boolean (opcional)

- Requiere: N/A
- Uso: oculta el archivo al usuario a menos que se produzca un error en la comprobación de validación. El valor predeterminado es true.
- Ejemplo - @hidden

@button boolean (opcional)

- Requiere: N/A
- Uso: la anotación debe estar en una propiedad booleana. Añade un botón que se sintetizará como verdadero cuando se elija. No es un botón.
- Ejemplo - buttonExample: boolean;

```
/**
 * @button
 */
buttonExample: boolean;
```

@showName boolean (opcional)

- Requiere: N/A
- Uso: solo se puede usar en un tipo de conexión de cuenta. Muestra la entrada de nombre oculta. El valor predeterminado es `default_environment`.
- Ejemplo - `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
  ...
}>;
```

@showEnvironmentType boolean (opcional)

- Requiere: N/A
- Uso: solo se puede usar en un tipo de conexión de cuenta. Muestra el menú desplegable de tipos de entornos ocultos. Todas las conexiones son de forma predeterminada. `production` Las opciones son Producción o No Producción.
- Ejemplo - `@showEnvironmentType true`

```
/**
 * @showEnvironmentType true
 */
accountConnection: AccountConnection<{
  ...
}>;
```

@forceDefault boolean (opcional)

- Requiere: N/A
- Uso: utiliza el valor predeterminado proporcionado por el autor del blueprint en lugar del valor que utilizaba anteriormente el usuario.
- Ejemplo: `forceDefaultExample: any;`

```
/**
```

```
* @forceDefault
*/
forceDefaultExample: any;
```

@requires blueprintName

- Requiere: anota la interfaz `Options`
- Uso: advierte al usuario que aplique lo especificado `blueprintName` al proyecto como requisito para el esquema actual.
- Ejemplo: `@requires '@amazon-codecatalyst/blueprints.blueprint-builder'`

```
/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
  ...
}
```

TypeScript Tipos compatibles

Los siguientes TypeScript tipos son compatibles con un esquema personalizado `Options` en el asistente de front-end.

Número

- Requiere: opción para ser un tipo. `number`
- Uso: genera un campo de entrada numérico.
- Ejemplo - `age: number`

```
{
  age: number
  ...
}
```

Cadena

- Requiere: opción para ser un tipo `string`.
- Uso: genera un campo de entrada de cadena.

- Ejemplo - name: string

```
{
  age: string
  ...
}
```

Lista de cadenas

- Requiere: opción para ser un tipo boolean.
- Uso: genera una casilla de verificación.
- Ejemplo - isProduction: boolean

```
{
  isProduction: boolean
  ...
}
```

Radio

- Requiere: la opción de ser una unión de tres o menos cuerdas.
- Uso: genera una radio seleccionada.

Note

Cuando hay cuatro o más elementos, este tipo se representa como un menú desplegable.

- Ejemplo: color: 'red' | 'blue' | 'green'

```
{
  color: 'red' | 'blue' | 'green'
  ...
}
```

Lista desplegable

- Requiere que la opción sea una unión de cuatro o más cadenas.

- **Uso:** genera un menú desplegable.
- **Ejemplo - runtimes:** 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'

```
{
  runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'
  ...
}
```

Sección ampliable

- **Requiere:** opción para ser un objeto.
- **Uso:** genera una sección ampliable. Las opciones del objeto se ubicarán dentro de la sección expandible del asistente.
- **Ejemplo:**

```
{
  expandableSectionTitle: {
    nestedString: string;
    nestedNumber: number;
  }
}
```

Tupla

- **Requiere:** la opción es ser del tipo `Tuple`.
- **Uso:** genera una entrada pagada con un valor clave.
- **Ejemplo - tuple:** `Tuple[string, string]>`

```
{
  tuple: Tuple[string, string]>;
  ...
}
```

Lista de tuplas

- **Requiere** que la opción sea un tipo `Tuple` de matriz.

- **Uso:** genera una entrada de lista de tuplas.
- **Ejemplo - tupleList:** `Tuple[string, string]>[]`

```
{  
  tupleList: Tuple[string, string]>[];  
  ...  
}
```

Selector

- **Requiere:** la opción debe ser del tipo `Selector`.
- **Uso:** genera un menú desplegable con los repositorios de origen o los planos aplicados a un proyecto.
- **Ejemplo:** `sourceRepo: Selector<SourceRepository>`

```
{  
  sourceRepo: Selector<SourceRepository>;  
  sourceRepoOrAdd: Selector<SourceRepository | string>;  
  blueprintInstantiation: Selector<BlueprintInstantiation>;  
  ...  
}
```

Selección múltiple

- **Requiere:** la opción debe ser del tipo `Selector`.
- **Uso:** genera una entrada de selección múltiple.
- **Ejemplo - multiselect:** `MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>`

```
{  
  multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>;  
  ...  
}
```

Comunicarse con el usuario durante la síntesis

Como autor de un plan, puede comunicarse con los usuarios más allá de los mensajes de validación. Por ejemplo, un miembro del espacio podría ver una combinación de opciones que

genere un plan que no esté claro. Los planos personalizados permiten comunicar los mensajes de error a los usuarios mediante la invocación de la síntesis. El esquema base implementa una `throwSynthesisError(...)` función que espera un mensaje de error claro. Puede invocar el mensaje mediante lo siguiente:

```
//blueprint.ts
this.throwSynthesisError({
  name: BlueprintSynthesisErrorTypes.BlueprintSynthesisError,
  message: 'hello from the blueprint! This is a custom error communicated to the
user.'
})
```

Componentes del entorno

El asistente de planos personalizados se genera dinámicamente desde la `Options` interfaz expuesta a través del asistente. Los planos permiten generar componentes de interfaz de usuario (UI) a partir de tipos expuestos.

Para importar los componentes del entorno de Amazon CodeCatalyst blueprints

En su `blueprint.ts` archivo, añada lo siguiente:

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

Temas

- [Creación de entornos de desarrollo](#)
- [Ejemplos de interfaces simuladas](#)

Creación de entornos de desarrollo

El siguiente ejemplo muestra cómo implementar la aplicación en la nube:

```
export interface Options extends ParentOptions {
  ...
  myNewEnvironment: EnvironmentDefinition{
    thisIsMyFirstAccountConnection: AccountConnection{
      thisIsARole: Role['lambda', 's3', 'dynamo'];
    };
  };
};
```



```
}

```

La interfaz genera un componente de interfaz de usuario que solicita un nuevo entorno (`myNewEnvironment`) con una conexión de cuenta única (`thisIsMyFirstAccountConnection`). También se genera un rol en la conexión de la cuenta (`thisIsARole`) con `['lambda', 's3', 'dynamo']` las capacidades de rol mínimas requeridas. No todos los usuarios tienen conexiones de cuenta, por lo que deberías comprobar si un usuario no conecta una cuenta o no conecta una cuenta con un rol. Los roles también se pueden anotar con `@inlinePolicies`. Para obtener más información, consulte [@inlinePolicy. /path/to/policy/file.json](#).

El componente de entorno requiere una `name` y `environmentType`. El siguiente código es la forma predeterminada mínima requerida:

```
{
  ...
  "myNewEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION"
  },
}
```

A continuación, el componente de interfaz de usuario le solicitará varios campos. A medida que rellena los campos, el plano adquiere una forma completamente expandida. Puede resultarle útil incluir el simulacro completo en el `defaults.json` archivo con fines de prueba y desarrollo.

Ejemplos de interfaces simuladas

Interfaz simulada simple

```
{
  ...
  "thisIsMyEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION",
    "thisIsMySecondAccountConnection": {
      "id": "12345678910",
      "name": "my-account-connection-name",
      "secondAdminRole": {
        "arn": "arn:aws:iam::12345678910:role/ConnectedQuokkaRole",
        "name": "ConnectedQuokkaRole",
        "capabilities": [
```

```
        "lambda",
        "s3",
        "dynamo"
    ]
}
}
```

Interfaz simulada compleja

```
export interface Options extends ParentOptions {
  /**
   * The name of an environment
   * @displayName This is a Environment Name
   * @collapsed
   */
  thisIsMyEnvironment: EnvironmentDefinition{
    /**
     * comments about the account that is being deployed into
     * @displayName This account connection has an overridden name
     * @collapsed
     */
    thisIsMyFirstAccountConnection: AccountConnection{
      /**
       * Blah blah some information about the role that I expect
       * e.g. here's a copy-pastable policy: [to a link]
       * @displayName This role has an overridden name
       */
      adminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
      /**
       * Blah blah some information about the second role that I expect
       * e.g. here's a copy-pastable policy: [to a link]
       */
      lambdaRole: Role['lambda', 's3'];
    };
    /**
     * comments about the account that is being deployed into
     */
    thisIsMySecondAccountConnection: AccountConnection{
      /**
       * Blah blah some information about the role that I expect
       * e.g. here's a copy-pastable policy: [to a link]
       */
    }
  }
}
```

```
    */
    secondAdminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
  /**
   * Blah blah some information about the second role that I expect
   * e.g. here's a copy-pastable policy: [to a link]
   */
    secondLambdaRole: Role['lambda', 's3'];
  };
};
}
```

Interfaz simulada completa

```
{
  ...
  "thisIsMyEnvironment": {
    "name": "my-production-environment",
    "environmentType": "PRODUCTION",
    "thisIsMySecondAccountConnection": {
      "id": "12345678910",
      "name": "my-connected-account",
      "secondAdminRole": {
        "name": "LambdaQuokkaRole",
        "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
        "capabilities": [
          "admin",
          "lambda",
          "s3",
          "cloudfront"
        ]
      },
    },
    "secondLambdaRole": {
      "name": "LambdaQuokkaRole",
      "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
      "capabilities": [
        "lambda",
        "s3"
      ]
    }
  },
  "thisIsMyFirstAccountConnection": {
    "id": "12345678910",
    "name": "my-connected-account",
```

```
"adminRole": {
  "name": "LambdaQuokkaRole",
  "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
  "capabilities": [
    "admin",
    "lambda",
    "s3",
    "cloudfront"
  ]
},
"lambdaRole": {
  "name": "LambdaQuokkaRole",
  "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
  "capabilities": [
    "lambda",
    "s3"
  ]
}
},
}
```

Componentes secretos

Los secretos se pueden usar CodeCatalyst para almacenar datos confidenciales a los que se puede hacer referencia en los flujos de trabajo. Puede añadir un secreto a su esquema personalizado y hacer referencia a él en su flujo de trabajo. Para obtener más información, consulte [Trabajando con secretos](#).

Para importar los CodeCatalyst blueprints de Amazon, escriba la región

En su `blueprint.ts` archivo, añada lo siguiente:

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-
component.secrets'
```

Temas

- [Crear un secreto](#)
- [Hacer referencia a un secreto en un flujo de trabajo](#)

Crear un secreto

En el siguiente ejemplo, se crea un componente de interfaz de usuario que solicita al usuario que introduzca un valor secreto y una descripción opcional:

```
export interface Options extends ParentOptions {
  ...
  mySecret: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
  constructor(options_: Options) {
    new Secret(this, options.secret);
  }
}
```

El componente secreto requiere unname. El siguiente código es la forma predeterminada mínima requerida:

```
{
  ...
  "secret": {
    "name": "secretName"
  },
}
```

Hacer referencia a un secreto en un flujo de trabajo

En el siguiente ejemplo, se crea un secreto y un flujo de trabajo que hace referencia al valor secreto. Para obtener más información, consulte [Hacer referencia a un secreto en un flujo de trabajo](#).

```
export interface Options extends ParentOptions {
  ...
  /**
   *
   * @validationRegex /^\\w+$/
   */
  username: string;
}
```

```
password: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
  constructor(options_: Options) {
    const password = new Secret(this, options_.password);

    const workflowBuilder = new WorkflowBuilder(this, {
      Name: 'my_workflow',
    });

    workflowBuilder.addAction({
      actionName: 'download_files',
      input: {
        Sources: ['WorkflowSource'],
      },
      output: {
        Artifacts: [{ Name: 'download', Files: ['file1'] }],
      },
      steps: [
        `curl -u ${options_.username}:${password.reference} https://example.com`,
      ],
    });

    new Workflow(
      this,
      repo,
      workflowBuilder.getDefinition(),
    );
  }
}
```

Para obtener más información sobre el uso de secretos en CodeCatalyst, consulte [Trabajando con secretos](#).

Componentes de la región

El tipo de región se puede añadir a la `Options` interfaz de su blueprint personalizado para generar un componente en el asistente de blueprint. Puede introducir uno o más giones de AWS. El tipo de región se puede importar desde el plano base al archivo. `blueprint.ts` Para obtener más información, consulte [las regiones de AWS](#).

Para importar los CodeCatalyst blueprints de Amazon, escriba la región

En su `blueprint.ts` archivo, añada lo siguiente:

```
import { Region } from '@amazon-codecatalyst/blueprints.blueprint'
```

El parámetro de tipo de región es un conjunto de códigos de región de AWS entre los que puede elegir o que puede utilizar `*` para incluir todas las regiones de AWS compatibles.

Temas

- [Annotations](#)
- [Ejemplos de componentes de región](#)

Annotations

Se pueden añadir etiquetas JSDoc a cada campo de la `Options` interfaz para personalizar el aspecto y el comportamiento de un campo en el asistente. Para el tipo de región, se admiten las siguientes etiquetas:

- La `@displayName` anotación se puede usar para cambiar la etiqueta del campo en el asistente.

Ejemplo: `@displayName AWS Region`

- La `@placeholder` anotación se puede utilizar para cambiar el marcador de posición del componente de selección o selección múltiple.

Ejemplo: `@placeholder Choose AWS Region`

Ejemplos de componentes de región

Elegir una región de una lista especificada

```
export interface Options extends ParentOptions {  
    ...  
    /**  
     * @displayName Region  
     */  
    region: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>;  
}
```

Elegir una o más regiones de una lista específica

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

Elegir una región de AWS

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Region
   */
  region: Region<['*']>;
}
```

Elegir una o más regiones de una lista específica

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

Componentes de repositorio y código fuente

Amazon utiliza un repositorio CodeCatalyst para almacenar el código. El repositorio toma un nombre como entrada. La mayoría de los componentes se almacenan en un repositorio, como los archivos de código fuente, los flujos de trabajo y otros componentes, como los entornos de desarrollo gestionados (MDE). El componente del repositorio de origen también exporta los componentes que se utilizan para administrar archivos y activos estáticos. Los repositorios tienen restricciones de nombre. Para obtener más información, consulte [Repositorios de fuentes en CodeCatalyst](#).

```
const repository = new SourceRepository(this, {
```



```
title: 'my-new-repository-title',
});
```

Para importar componentes de código fuente y repositorio de CodeCatalyst blueprints de Amazon

En su `blueprint.ts` archivo, añade lo siguiente:

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

Temas

- [Añadir un archivo](#)
- [Añadir un archivo genérico](#)
- [Copiar archivos](#)
- [Dirigirse a varios archivos](#)
- [Crear un repositorio nuevo y añadir archivos](#)

Añadir un archivo

Puede escribir un archivo de texto en un repositorio con la `SourceFile` construcción. La operación es uno de los casos de uso más comunes y requiere un repositorio, una ruta de archivo y contenido de texto. Si la ruta del archivo no existe en un repositorio, el componente crea todas las carpetas necesarias.

```
new SourceFile(repository, `path/to/my/file/in/repo/file.txt`, 'my file contents');
```

Note

Si escribe dos archivos en la misma ubicación dentro del mismo repositorio, la implementación más reciente sobrescribe la anterior. Puede utilizar la función para superponer el código generado y es especialmente útil para ampliar el código que puedan haber generado los planos personalizados.

Añadir un archivo genérico

Puede escribir bits arbitrarios en su repositorio. Puedes leer desde un búfer y usar la `File` construcción.

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

Copiar archivos

Puede empezar con el código generado copiando y pegando el código de inicio y, a continuación, generando más código sobre esa base. Coloca el código dentro del `static-assets` directorio y, a continuación, apunta a ese código con la `StaticAsset` construcción. En este caso, la ruta siempre comienza en la raíz del `static-assets` directorio.

```
const starterCode = new StaticAsset('path/to/file/file.txt')
const starterCodeText = new StaticAsset('path/to/file/file.txt').toString()
const starterCodeRawContent = new StaticAsset('path/to/image/hello.png').content()

const starterCodePath = new StaticAsset('path/to/image/hello.png').path()
// starterCodePath is equal to 'path/to/image/hello.png'
```

Una subclase de `StaticAsset` es `SubstitutionAsset`. La subclase funciona exactamente igual, pero en su lugar puedes sustituir el bigote por encima del archivo. Puede resultar útil para realizar la generación de `copy-and-replace` estilos.

La sustitución de activos estáticos utiliza un motor de plantillas `Moustache` para renderizar los archivos estáticos que se incluyen en el repositorio fuente generado. Las reglas de creación de plantillas de `Moustache` se aplican durante la renderización, lo que significa que todos los valores están codificados en HTML de forma predeterminada. Para renderizar HTML sin escapes, utilice la sintaxis de triple bigote. `{{{name}}}` Para obtener más información, consulta las reglas de creación de plantillas para [bigotes](#).

Note

Si se utiliza un sustituto sobre archivos que no son interpretables por texto, se pueden producir errores.

```
const starterCodeText = new SubstitutionAsset('path/to/file/file.txt').subsitite({
  'my_variable': 'subbed value1',
  'another_variable': 'subbed value2'
```

```
})
```

Dirigirse a varios archivos

Los activos estáticos admiten la segmentación global mediante una función estática activada `StaticAsset` y sus subclases llamadas `findAll(...)`, que devuelve una lista de los activos estáticos precargados con sus rutas, contenidos y mucho más. Puede encadenar la lista con `File` construcciones para copiar y pegar el contenido en el directorio. `static-assets`

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

Crear un repositorio nuevo y añadir archivos

Puede usar un componente de repositorio para crear un repositorio nuevo en un proyecto generado. A continuación, puede añadir archivos o flujos de trabajo al repositorio creado.

```
import { SourceRepository } from '@amazon-codecatalyst/codecatalyst-source-
repositories';
...
const repository = new SourceRepository(this, { title: 'myRepo' });
```

El siguiente ejemplo muestra cómo añadir archivos y flujos de trabajo a un repositorio existente:

```
import { SourceFile } from '@amazon-codecatalyst/codecatalyst-source-repositories';
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows';
...
new SourceFile(repository, 'README.md', 'This is the content of my readme');
new Workflow(this, repository, {/**...workflowDefinition...**/});
```

La combinación de los dos fragmentos de código genera un único repositorio denominado `myRepo` con un archivo fuente `README.md` y un CodeCatalyst flujo de trabajo en la raíz.

Componentes del flujo de trabajo

Los CodeCatalyst proyectos de Amazon utilizan un flujo de trabajo para ejecutar acciones en función de los desencadenantes. Puedes usar los componentes del flujo de trabajo para crear y armar archivos YAML del flujo de trabajo. Para obtener más información, consulte [Referencia de definición de flujo de trabajo](#).

Para importar componentes de flujos de trabajo de Amazon CodeCatalyst blueprints

En su `blueprint.ts` archivo, añada lo siguiente:

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

Temas

- [Ejemplos de componentes del flujo de trabajo](#)
- [Conectarse a un entorno](#)

Ejemplos de componentes del flujo de trabajo

WorkflowBuilder componente

Puede usar una clase para crear una definición de flujo de trabajo. La definición se puede asignar a un componente del flujo de trabajo para renderizarlo en un repositorio.

```
import { WorkflowBuilder } from '@amazon-codecatalyst/codecatalyst-workflows'

const workflowBuilder = new WorkflowBuilder({} as Blueprint, {
  Name: 'my_workflow',
});

// trigger the workflow on pushes to branch 'main'
workflowBuilder.addBranchTrigger(['main']);

// add a build action
workflowBuilder.addAction({
  // give the action a name
  actionName: 'build_and_do_some_other_stuff',

  // the action pulls from source code
  input: {
    Sources: ['WorkflowSource'],
  },

  // the output attempts to autodiscover test reports, but not in the node modules
  output: {
    AutoDiscoverReports: {
      Enabled: true,
      ReportNamePrefix: AutoDiscovered,
    }
  }
});
```

```
    IncludePaths: ['**/*'],
    ExcludePaths: ['*/node_modules/**/*'],
  },
},
// execute some arbitrary steps
steps: [
  'npm install',
  'npm run myscript',
  'echo hello-world',
],
// add an account connection to the workflow
environment: convertToWorkflowEnvironment(myEnv),
});
```

Componente del proyecto de flujo de trabajo

El siguiente ejemplo muestra cómo se puede usar un componente de Projen para escribir un flujo de trabajo YAML en un repositorio:

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'

...

const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// can also pass in any object and have it rendered as a yaml. This is unsafe and may
not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

Conectarse a un entorno

Muchos flujos de trabajo deben ejecutarse en una conexión de cuenta de AWS. Los flujos de trabajo resuelven este problema al permitir que las acciones se conecten a entornos con especificaciones de nombre de cuenta y rol.

```
import { convertToWorkflowEnvironment } from '@amazon-codecatalyst/codecatalyst-workflows'
```

```
const myEnv = new Environment(...);

// can be passed into a workflow constructor
const workflowEnvironment = convertToWorkflowEnvironment(myEnv);

// add a build action
workflowBuilder.addAction({
  ...
  // add an account connection to the workflow
  environment: convertToWorkflowEnvironment(myEnv),
});
```

Componentes de entornos de desarrollo

Los entornos de desarrollo gestionados (MDE) se utilizan para crear y poner en marcha espacios de trabajo MDE. CodeCatalyst El componente genera un archivo `devfile.yaml` Para obtener más información, consulte [Introducción a Devfile](#) y [Mover un archivo de desarrollo del repositorio a un entorno de desarrollo](#).

```
new Workspace(this, repository, SampleWorkspaces.default);
```

Para importar los componentes de los espacios de trabajo de Amazon CodeCatalyst blueprints

En su `blueprint.ts` archivo, añada lo siguiente:

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

Componentes de los problemas

En CodeCatalyst, puedes monitorear las características, las tareas, los errores y cualquier otro trabajo relacionado con tu proyecto. Cada pieza de trabajo se guarda en un registro distinto, denominado edición. Cada número puede tener una descripción, un responsable, un estado y otras propiedades, que puede buscar, agrupar y filtrar. Puede ver sus problemas con las vistas predeterminadas o puede crear las suyas propias con filtros, clasificaciones o agrupaciones personalizados. Para obtener más información sobre los conceptos relacionados con los problemas, consulte [Conceptos de problemas](#) y [Cuotas para emisiones en CodeCatalyst](#).

El componente de problema genera una representación en JSON de un problema. El componente incluye un campo de ID y una definición del problema como entrada.

Para importar componentes de problemas de CodeCatalyst blueprints de Amazon

En tu `blueprint.ts` archivo, añade lo siguiente:

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

Temas

- [Emite, ejemplos de componentes](#)

Emite, ejemplos de componentes

Creando un problema

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
  title: 'myFirstIssue',
  content: 'This is an example issue.',
});
```

Crear un problema de alta prioridad

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
...
const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// Creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// Can also pass in any object and have it rendered as a yaml. This is unsafe and may
  not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

Crear un problema de baja prioridad con etiquetas

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myThirdIssue', {
  title: 'myThirdIssue',
```

```
content: 'This is an example of a low priority issue with a label.',
priority: 'LOW',
labels: ['exampleLabel'],
});
```

Trabajar con herramientas de planos y CLI

La [CLI de blueprint](#) proporciona herramientas para administrar sus planos personalizados y trabajar con ellos.

Temas

- [Trabajando con herramientas de blueprint](#)
- [Herramienta de carga de imágenes](#)

Trabajando con herramientas de blueprint

Para trabajar con las herramientas de planos

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Reanude su entorno de desarrollo. Para obtener más información, consulte [Reanudación de un entorno de desarrollo](#).

Si no tiene un entorno de desarrollo, primero debe crear uno. Para obtener más información, consulte [Creación de un entorno de desarrollo](#).

3. En una terminal que funcione, ejecute el siguiente comando para instalar la CLI del blueprint:

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. En el `blueprint.ts` archivo, importe las herramientas que desee utilizar en el siguiente formato:

```
import { <tooling-function-name> } from '@amazon-codecatalyst/blueprint-util.cli/lib/<tooling-folder-name>/<tooling-file-name>;
```

Tip

Puede ir al [CodeCatalyst blueprints GitHub repository](#) para buscar el nombre de la herramienta que desea utilizar.

Si desea utilizar la herramienta de carga de imágenes, añada lo siguiente a su script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

Ejemplos

- Si desea utilizar la función de publicación, añada lo siguiente a su script:

```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/  
publish';
```

- Si desea utilizar la herramienta de carga de imágenes, añada lo siguiente a su script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

5. Llama a la función.

Ejemplos:

- Si desea utilizar la función de publicación, añada lo siguiente a su script:

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Si desea utilizar la herramienta de carga de imágenes, añada lo siguiente a su script:

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/  
image');
```

Herramienta de carga de imágenes

La herramienta de carga de imágenes le permite cargar su propia imagen en un bucket de S3 de su cuenta de AWS y luego distribuirla públicamente detrás CloudFront. La herramienta toma como entrada una ruta de imagen en el almacenamiento local (y el nombre del depósito opcional) y devuelve la URL de la imagen que está disponible públicamente. Para obtener más información, consulta [¿Qué es Amazon CloudFront?](#) y [¿Qué es Amazon S3?](#)

Para trabajar con la herramienta de carga de imágenes

1. Clone el [GitHub repositorio de blueprints de código abierto](#) que proporciona acceso al SDK de blueprints y a los blueprints de muestra. En una terminal que funcione, ejecute el siguiente comando:

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Ejecute el siguiente comando para ir al GitHub repositorio de planos:

```
cd codecatalyst-blueprints
```

3. Ejecute el siguiente comando para instalar las dependencias:

```
yarn && yarn build
```

4. Ejecute el siguiente comando para asegurarse de que está instalada la última versión de la CLI del blueprint:

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Inicie sesión en la cuenta de AWS con el bucket de S3 en el que desee cargar la imagen. Para obtener más información, consulte [Configurar la CLI de AWS](#) e [iniciar sesión a través de la interfaz de línea de comandos de AWS](#).

6. Ejecute el siguiente comando desde la raíz del CodeCatalyst repositorio para navegar al directorio con la CLI del blueprint:

```
cd packages/utils/blueprint-cli
```

7. Ejecute el siguiente comando para cargar la imagen en un bucket de S3:

```
yarn blueprint upload-image-public <./path/to/your/image>  
  <optional:optional-bucket-name>
```

Se generará la URL de la imagen. La URL no estará disponible de forma inmediata, ya que la CloudFront distribución tardará algún tiempo en implementarse. Compruebe el estado de la distribución para obtener el estado más reciente de la implementación. Para obtener más información, consulte [Trabajar con distribuciones](#).

Pruebas de instantáneas

Se admiten las pruebas de instantáneas generadas en varias configuraciones de su blueprint.

Los blueprints permiten realizar [pruebas instantáneas](#) en las configuraciones que usted haya proporcionado como autor del blueprint. Las configuraciones son anulaciones parciales que se combinan en la parte superior del archivo `defaults.json`, en la raíz de un blueprint. Cuando las pruebas de instantáneas están habilitadas y configuradas, el proceso de compilación y prueba sintetiza las configuraciones dadas y verifica que los resultados sintetizados no hayan cambiado con respecto a la instantánea de referencia. Para ver el código de prueba de las instantáneas, consulta el repositorio de [CodeCatalyst planos GitHub](#).

Para habilitar las pruebas de instantáneas

1. En el `.projenrc.ts` archivo, actualice el objeto de entrada `ProjenBlueprint` con los archivos que desee capturar. Por ejemplo:

```
{
  ....
  blueprintSnapshotConfiguration: {
    snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],
  },
}
```

2. Vuelva a sintetizar el blueprint para crear TypeScript archivos en su proyecto de blueprint. No edite los archivos fuente, ya que Projen los mantiene y los regenera. Utilice el siguiente comando:

```
yarn projen
```

3. Navegue hasta el `src/snapshot-configurations` directorio para ver el `default-config.json` archivo con un objeto vacío. Actualice o sustituya el archivo por una o más de sus propias configuraciones de prueba. A continuación, cada configuración de prueba se fusiona con el `defaults.json` archivo del proyecto, se sintetiza y se compara con las instantáneas durante las pruebas. Use el siguiente comando para realizar la prueba:

```
yarn test
```

La primera vez que utilice un comando de prueba, aparecerá el siguiente mensaje: `Snapshot Summary > NN snapshots written from 1 test suite`. Las pruebas posteriores

verifican que la salida sintetizada no haya cambiado con respecto a las instantáneas y muestran el siguiente mensaje: `Snapshots: NN passed, NN total`.

Si cambias el plano de forma intencionada para producir un resultado diferente, ejecuta el siguiente comando para actualizar las instantáneas de referencia:

```
yarn test:update
```

Las instantáneas esperan que los resultados sintetizados sean constantes entre cada ejecución. Si el blueprint genera archivos que varían, debe excluirlos de las pruebas de instantáneas. Actualice el `blueprintSnapshotConfiguration` objeto del objeto `ProjenBlueprint` de entrada para añadir la `snapshotGlobs` propiedad. La `snapshotGlobs` propiedad es una matriz de [globos](#) que determina qué archivos se incluyen o excluyen de la captura de pantalla.

Note

Hay una lista predeterminada de globos. Si especificas tu propia lista, es posible que tengas que recuperar de forma explícita las entradas predeterminadas.

Publicar un blueprint personalizado

Los planos personalizados proporcionan paquetes de vistas previas como resultado de una síntesis exitosa. El paquete de proyectos representa el código fuente, la configuración y los recursos de un proyecto, y las operaciones de la API de CodeCatalyst despliegue lo utilizan para implementarlo en un proyecto. Si quieres seguir desarrollando tu plan personalizado, vuelve a ejecutar el proceso de síntesis del modelo. Para obtener más información, consulte [Conceptos de planos personalizados](#).

Important

Si desea utilizar paquetes de planos de fuentes externas, tenga en cuenta los riesgos que pueden conllevar dichos paquetes. Eres responsable de los planos personalizados que añadas a tu espacio y del código que generen.

Temas

- [Ver y publicar una versión preliminar de un plano personalizado](#)

- [Ver y publicar una versión normal de un esquema personalizado](#)
- [Publicar y aplicar un plano personalizado en espacios y proyectos específicos](#)

Ver y publicar una versión preliminar de un plano personalizado

Puedes publicar una versión preliminar de tu plano personalizado en tu espacio si quieres añadirlo al catálogo de planos de tu espacio. Esto le permite ver el plano como usuario antes de añadir la versión sin vista previa al catálogo. La versión preliminar le permite publicar sin tener que utilizar una versión real. Por ejemplo, si trabaja en una `0.0.1` versión, puede publicar y añadir una versión preliminar, de forma que las nuevas actualizaciones de una segunda versión puedan publicarse y añadirse como tal `0.0.2`.

Tras realizar los cambios, reconstruya el paquete del blueprint personalizado ejecutando el `package.json` archivo y previsualice los cambios.

Para ver y publicar una versión preliminar de un blueprint personalizado

1. Reanude su entorno de desarrollo. Para obtener más información, consulte [Reanudación de un entorno de desarrollo](#)
2. Abra una terminal de trabajo en su entorno de desarrollo.
3. (Opcional) En una terminal de trabajo, instala las dependencias necesarias para tu proyecto si aún no las has instalado. Utilice el siguiente comando:

```
yarn
```

4. (Opcional) Si has realizado cambios en el `.projenrc.ts` archivo, vuelve a generar la configuración del proyecto antes de crear y previsualizar el blueprint. Utilice el siguiente comando:

```
yarn projen
```

5. Reconstruya y obtenga una vista previa del plano personalizado mediante el siguiente comando. Utilice el siguiente comando:

```
yarn blueprint:preview
```

Navegue hasta el `See this blueprint at:` enlace proporcionado para obtener una vista previa del plano personalizado. Comprueba que la interfaz de usuario, incluido el texto,

se muestre como esperabas en función de tu configuración. Si desea cambiar el esquema personalizado, puede editar el `blueprint.ts` archivo, volver a sintetizar el esquema y, a continuación, volver a publicar una versión preliminar. Para obtener más información, consulte [Resíntesis](#).

6. (Opcional) Puedes publicar una versión preliminar de tu plano personalizado, que luego podrás añadir al catálogo de planos de tu espacio. Navega hasta el `Enable version [preview version number]` at : enlace para publicar una versión preliminar en tu espacio.

Puedes emular la creación de proyectos sin tener que crear un proyecto en CodeCatalyst él. Para sintetizar tu proyecto, usa el siguiente comando:

```
yarn blueprint:synth
```

Se genera un plano en la `synth/synth.[options-name]/proposed-bundle/` carpeta. Para obtener más información, consulte [Síntesis](#).

Si va a actualizar su esquema personalizado, utilice el siguiente comando para volver a sintetizar el proyecto:

```
yarn blueprint:resynth
```

Se genera un plano en la carpeta. `synth/synth.[options-name]/proposed-bundle/` Para obtener más información, consulte [Resíntesis](#).

Tras publicar la versión preliminar, puede añadir el plano para que los miembros del espacio puedan utilizarlo para crear nuevos proyectos o aplicarlo a proyectos existentes. Para obtener más información, consulte [Añadir un plano personalizado a un catálogo de espacios](#).

Ver y publicar una versión normal de un esquema personalizado

Cuando hayas terminado de desarrollar y previsualizar tu plano personalizado, podrás ver y publicar la nueva versión que quieras añadir al catálogo de planos de tu espacio. El flujo de trabajo de publicación que se genera al crear un proyecto publica automáticamente los cambios que se van introduciendo. Si desactivaste la generación del flujo de trabajo al crear el plano, este no estará disponible automáticamente para añadirlo al catálogo de planos de tu espacio. Puedes seguir publicando tu plano personalizado en tu espacio después de ejecutar un comando. `yarn`

Para ver y publicar un plano personalizado

1. Reanude su entorno de desarrollo. Para obtener más información, consulte [Reanudación de un entorno de desarrollo](#)
2. Abra una terminal de trabajo en su entorno de desarrollo.
3.
 - Si optó por no generar el flujo de trabajo de lanzamiento al crear su blueprint, utilice el siguiente comando:

```
yarn blueprint:release
```

Aún puede navegar hasta el `See this blueprint at:` enlace proporcionado para ver su plano personalizado.

Publica la versión actualizada de tu plano personalizado, que luego podrás añadir al catálogo de planos de tu espacio. Navega hasta el `Enable version [release version number] at:` enlace para publicar la versión más reciente en tu espacio.

- Si optó por un flujo de trabajo de publicación al crear su blueprint, la última versión del blueprint se publicará automáticamente cuando se introduzcan cambios. Utilice los siguientes comandos:

```
git add .
```

```
git commit -m "commit message"
```

```
git push
```

Tras publicar tu versión normal, puedes añadir el esquema para que los miembros del espacio puedan usarlo para crear nuevos proyectos o aplicarlo a proyectos existentes. Para obtener más información, consulte [Añadir un plano personalizado a un catálogo de espacios](#).

Publicar y aplicar un plano personalizado en espacios y proyectos específicos

De forma predeterminada, los `blueprint:release` comandos `blueprint:preview` y se publican en el CodeCatalyst espacio en el que se creó el plano. Si tiene varios espacios empresariales, también puede previsualizar y publicar el mismo plano en esos espacios. También puedes aplicar un plano a un proyecto existente de otro espacio.

Para publicar o aplicar un plano personalizado en un espacio específico

1. Reanude su entorno de desarrollo. Para obtener más información, consulte [Reanudación de un entorno de desarrollo](#).
2. Abra una terminal de trabajo en su entorno de desarrollo.
3. (Opcional) Instala las dependencias necesarias para tu proyecto si aún no las has instalado. Utilice el siguiente comando:

```
yarn
```

4. Usa la `--space` etiqueta para publicar una vista previa o una versión normal en un espacio específico. Por ejemplo:

- ```
yarn blueprint:preview --space my-awesome-space # publishes under a "preview" version tag to 'my-awesome-space'
```

Ejemplo de salida:

```
Enable version 0.0.1-preview.0 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.0/projects/create
```

- ```
yarn blueprint:release --space my-awesome-space # publishes normal version to 'my-awesome-space'
```

Ejemplo de salida:

```
Enable version 0.0.1 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1/projects/create
```

Utilice la `--project` para aplicar una versión preliminar de un esquema personalizado a un proyecto existente en un espacio específico. Por ejemplo:


```
yarn blueprint:preview --space my-awesome-space --project my-project # previews
blueprint application to an existing project
```

Ejemplo de salida:

```
Enable version 0.0.1-preview.1 at: https://codecatalyst.aws/spaces/my-awesome-
space/blueprints
Blueprint applied to [my-project]: https://codecatalyst.aws/spaces/my-awesome-
space/projects/my-project/blueprints/%40amazon-codecatalyst%2FmySpace.my-blueprint/
publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.1/add
```

Visualización de los detalles, las versiones y los proyectos de un blueprint personalizado

Puedes ver los planos personalizados publicados de tu espacio, incluidos los detalles del plano, las versiones y los proyectos que se crean con el plano o se aplican a él.

Temas

- [Ver los planos personalizados de un espacio](#)
- [Ver los proyectos creados con un plano personalizado o aplicarlos](#)

Ver los planos personalizados de un espacio

Para ver los planos personalizados de un espacio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee ver un plano personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos para ver los planos del espacio. En la tabla se muestran los siguientes detalles:
 - Nombre: nombre del plano personalizado.
 - Estado del catálogo: si el plano personalizado se ha publicado en el catálogo de planos del espacio.

- Última versión: la versión más reciente del plano personalizado.
- Última modificación: fecha en la que se actualizó por última vez el plano del espacio.

Ver los proyectos creados con un plano personalizado o aplicarlos

Para ver los proyectos creados con un blueprint personalizado o a los que se ha aplicado

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee ver un plano personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. En la tabla de planos de espacios, elija el nombre de un plano personalizado para ver los proyectos que utilizan tablas de planos y los proyectos que no utilizan tablas de planos.

Añadir y eliminar un plano personalizado en un espacio

Después de publicar un plano personalizado en tu espacio, puedes añadirlo al catálogo de planos de tu espacio. También puedes eliminar un plano personalizado del catálogo de planos de tu espacio si ya no quieres que se utilice para crear nuevos proyectos o que se aplique a proyectos existentes.

Temas

- [Añadir un plano personalizado a un catálogo de espacios](#)
- [Eliminar un plano personalizado de un catálogo de espacios](#)

Añadir un plano personalizado a un catálogo de espacios

Si agregas un plano personalizado al catálogo de planos de tu CodeCatalyst espacio, todos los miembros del espacio podrán usarlo al crear un proyecto o aplicarlo a un proyecto existente. Antes de añadir un plano personalizado al catálogo de planos del espacio, debe estar habilitado el permiso de publicación del plano. Si optó por la generación de versiones del flujo de trabajo, los permisos de publicación están habilitados de forma predeterminada. Para obtener más información, consulte [Administrar los permisos de publicación de un blueprint personalizado](#) y [Publicar un blueprint personalizado](#).

Para añadir un plano al catálogo de planos del espacio

1. [Abra la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. El blueprint solo se puede añadir desde la rama predeterminada del repositorio de origen. Si ha desarrollado el plano en una rama de entidades, fusione su rama de entidades con los cambios de la rama predeterminada. Crea una solicitud de extracción para combinar cualquier cambio en la rama predeterminada. Para obtener más información, consulte [Trabajar con solicitudes de cambios en Amazon CodeCatalyst](#).
3. En la CodeCatalyst consola, navega hasta el panel de control del espacio con tu plano personalizado.
4. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
5. Elija el nombre del plano que desee añadir y, a continuación, seleccione Añadir al catálogo. Si tiene más de una versión, elija una en el menú desplegable de la versión del catálogo.
6. Seleccione Guardar.

Eliminar un plano personalizado de un catálogo de espacios

Note

Si elimina un plano personalizado de un catálogo de espacios, no afectará a los proyectos creados a partir del plano ni a los proyectos en los que se haya aplicado el plano. Los recursos del plano no se eliminan del proyecto.

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navega hasta el panel de control del espacio con tu plano personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. Elija el nombre del plano que desee eliminar y, a continuación, elija Eliminar el plano del catálogo.

Administrar los permisos de publicación de un blueprint personalizado

De forma predeterminada, el permiso de un blueprint personalizado está habilitado si se generó una versión del flujo de trabajo durante la creación del proyecto. Cuando los permisos de publicación

están habilitados, el blueprint se puede publicar en el espacio. Puede deshabilitar el permiso para que el blueprint no se pueda publicar. Cuando el permiso está deshabilitado, el flujo de trabajo de publicación que se genera durante la creación del blueprint no se ejecuta. Los cambios nuevos en un blueprint no se pueden publicar a menos que los permisos del blueprint estén habilitados.

 Important

Para habilitar o deshabilitar los permisos de publicación de un proyecto de blueprint, debe tener el rol de administrador del espacio.

Para gestionar los permisos de publicación de un proyecto de blueprint

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee gestionar los permisos de publicación de un blueprint personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. Selecciona la pestaña Permisos de publicación del proyecto para ver los permisos de publicación de todos los planos de tu espacio.
5. Elija el plano que desee administrar y, a continuación, seleccione Activar o Desactivar para cambiar los permisos de publicación. Si está habilitando los permisos, revise los detalles del cambio de permiso y, a continuación, seleccione Habilitar la publicación de planos para confirmar el cambio.

Administración de versiones para un blueprint personalizado

Como autor de planos, puedes administrar la versión que deseas publicar en el catálogo de planos del espacio. Cambiar la versión del catálogo de un plano no afecta a los proyectos que utilizan una versión de plano diferente.

Para administrar una versión de esquema personalizada

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee cambiar la versión de un blueprint personalizado.

3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. En la tabla de planos del espacio, selecciona el botón de opción correspondiente al plano personalizado que quieres gestionar.
5. Seleccione Crear versión de catálogo y, a continuación, elija una versión en el menú desplegable Versión de catálogo.
6. Seleccione Guardar.

Eliminar un blueprint o una versión personalizados publicados

Cuando eliminas la versión de un blueprint personalizado o el propio blueprint de tu CodeCatalyst espacio de Amazon, se elimina todo tu acceso a los recursos del proyecto del blueprint o de la versión del blueprint. Si eliminas una versión del plano o el plano, los miembros del proyecto no podrán acceder a los recursos del proyecto y se detendrán todos los flujos de trabajo que soliciten repositorios de fuentes de terceros.

Note

Si eliminas un esquema, esto no afectará al proyecto al que se haya aplicado el esquema. Los recursos del esquema no se eliminan del proyecto.

Si se publica una versión del plano en el catálogo de planos del espacio, elija una nueva versión para el catálogo antes de eliminar la versión publicada.

Para eliminar la versión del catálogo de un plano personalizado

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee eliminar la versión del catálogo de un blueprint personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. Elija el nombre del plano con la versión del catálogo que desee eliminar.
5. Elija el botón de opción de la versión del catálogo que desee eliminar y, a continuación, elija Eliminar versión.

6. Revise los detalles y, a continuación, elija otra versión del blueprint en el menú desplegable Elija una nueva versión del catálogo del blueprint.
7. Ingrese `delete` para confirmar la selección de la versión del catálogo de planos.
8. Elija Eliminar.

Si una versión de anteproyecto no está en el catálogo de planos del espacio, puede eliminarla sin tener que elegir una nueva.

Para eliminar una versión de plano personalizada

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee eliminar una versión de blueprint personalizada.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. Elija el nombre del plano con la versión que desee eliminar.
5. Elija el botón de radio de la versión que desee eliminar y, a continuación, elija Eliminar versión.
6. Introduzca este `delete` botón para confirmar la eliminación de la versión del blueprint.
7. Elija Eliminar.

Al eliminar un plano del catálogo de planos del espacio, se eliminan todas las versiones del plano. Los proyectos del espacio que utilizan el plano no se ven afectados por la eliminación.

Para eliminar una versión de plano personalizada

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la CodeCatalyst consola, navegue hasta el espacio en el que desee eliminar un blueprint personalizado.
3. En el panel de control del espacio, selecciona la pestaña Configuración y, a continuación, selecciona Planos.
4. En la tabla de planos del espacio, pulse el botón de opción del plano personalizado que desee eliminar y, a continuación, seleccione Eliminar plano.
5. Pulse aquí `delete` para confirmar la eliminación del plano personalizado.
6. Elija Eliminar.

Trabajar con dependencias y herramientas

Temas

- [Añadir dependencias](#)
- [Manejo de las discrepancias de los tipos de dependencia](#)
- [Usando yarn y npm](#)
- [Actualización de herramientas y componentes](#)

Añadir dependencias

Como autor de un blueprint, puede que necesite añadir paquetes al blueprint, por ejemplo. `@amazon-codecatalyst/blueprint-component.environments` [Debe actualizar el `projen.ts` archivo con ese paquete y, a continuación, volver a generar la configuración del proyecto con Projen](#). Projen actúa como modelo de proyecto para cada código base de planos, lo que permite actualizar las herramientas compatibles con versiones anteriores al cambiar la forma en que el modelo representa los archivos de configuración. El `package.json` archivo es un archivo que pertenece parcialmente al modelo de Projen. Projen reconoce las versiones dependientes incluidas en el archivo `package.json`, pero las demás opciones deben provenir del modelo.

Para añadir una dependencia y actualizar un archivo **`projenrc.ts`**

1. En el `projen.ts` archivo, navegue hasta la sección `deps`.
2. Añada la dependencia que desee usar en su blueprint.
3. Use el siguiente comando para regenerar la configuración de su proyecto:

```
yarn projen && yarn
```

Manejo de las discrepancias de los tipos de dependencia

Tras una actualización [de Yarn](#), es posible que aparezca el siguiente error relacionado con un parámetro del repositorio:

```
Type 'SourceRepository' is missing the following properties from type 'SourceRepository': synthesisSteps, addSynthesisStep
```

El error se debe a una falta de coincidencia de dependencias que se produce cuando un componente se basa en una versión más reciente de otro componente, pero el componente de confianza está anclado a una versión anterior. El error se puede corregir haciendo que todos los componentes dependan de la misma versión para que la versión se sincronice entre ellos. Es mejor mantener todos los paquetes vendidos por blueprint en la misma versión más reciente ($0.0.x$), a menos que estés seguro de cómo estás gestionando las versiones. El siguiente ejemplo muestra cómo se puede configurar el `package.json` archivo para que todas las dependencias dependan de la misma versión:

```
...
"@caws-blueprint-component/caws-environments": "^0.1.12345",
"@caws-blueprint-component/caws-source-repositories": "^0.1.12345",
"@caws-blueprint-component/caws-workflows": "^0.1.12345",
"@caws-blueprint-component/caws-workspaces": "^0.1.12345",
"@caws-blueprint-util/blueprint-utils": "^0.1.12345",
...
"@caws-blueprint/blueprints.blueprint": "*",
```

Tras configurar las versiones de todas las dependencias, utilice el siguiente comando:

```
yarn install
```

Usando yarn y npm

Los planos usan [hilo](#) como herramienta. El uso de [npm](#) y Yarn provocará problemas con las herramientas porque la forma en que cada uno resuelve los árboles de dependencias es diferente. Para evitar estos problemas, es mejor usar solo Yarn.

Si instalaste dependencias accidentalmente con npm, puedes eliminar el `package-lock.json` archivo generado y asegurarte de que esté actualizado con las dependencias que necesitas. `.projenrc.ts` Puedes regenerar la configuración de tu proyecto con Projen.

Utilice lo siguiente para regenerar a partir del modelo:

```
yarn projen
```

Tras comprobar que el archivo `.projenrc.ts` está actualizado con las dependencias necesarias, utilice el siguiente comando:


```
yarn
```

Actualización de herramientas y componentes

De vez en cuando, es posible que desee actualizar sus herramientas y componentes para incorporar las nuevas funciones disponibles. Se recomienda mantener todos los componentes en la misma versión, a menos que esté seguro de cómo va a gestionar las versiones. Las versiones se sincronizan entre los componentes, por lo que las mismas versiones para todos los componentes garantizan una dependencia adecuada entre ellos.

Uso de Yarn workspace monorepo

Usa el siguiente comando para actualizar las utilidades y los componentes desde la raíz del repositorio de un blueprint personalizado:

```
yarn upgrade @amazon-codecatalyst/*
```

Utilice el siguiente comando si no está utilizando un monorepo:

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Otras opciones que puede utilizar para actualizar las herramientas y los componentes:

- Use `npm view @caws-blueprint-component/<some-component>` para obtener la versión más reciente.
- Aumente manualmente a la última versión configurando la versión en su archivo `package.json` y utilizando el siguiente comando: `yarn`. Todos los componentes y utilidades deben tener la misma versión.

Contribuye

El kit de desarrollo de software (SDK) de blueprints es una biblioteca de código abierto a la que puedes contribuir. Como colaborador, ten en cuenta las pautas de la contribución, los comentarios y los defectos. Para obtener más información, consulta el [GitHub repositorio de planos](#).

Cuotas de planos en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de los blueprints en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Número máximo de planos aplicados por CodeCatalyst proyecto	100
---	-----

Repositorios de fuentes en CodeCatalyst

CodeCatalyst Los repositorios de código fuente son repositorios de Git alojados en Amazon.

CodeCatalyst Puedes usar los repositorios de código fuente CodeCatalyst para almacenar, versionar y gestionar de forma segura los activos de un proyecto.

Los activos de un CodeCatalyst repositorio pueden incluir:

- Documentos de
- código fuente
- archivos binarios

CodeCatalyst también utiliza el repositorio de origen de un proyecto para almacenar la información de configuración del proyecto, como los archivos de configuración del flujo de trabajo.

Puede tener más de un repositorio de origen en un CodeCatalyst proyecto. Por ejemplo, es posible que desees tener repositorios de fuentes independientes para el código fuente del front-end, el código fuente del back-end, las utilidades y la documentación.

Este es un posible flujo de trabajo para trabajar con código en repositorios de código fuente, solicitudes de cambios y entornos de desarrollo en: CodeCatalyst

Mary Major crea un proyecto de aplicación web CodeCatalyst utilizando un plano, que crea un repositorio de código fuente con código de muestra. Invita a sus amigos Li Juan, Saanvi Sarkar y Jorge Souza a trabajar en el proyecto con ella. Li Juan examina el código de muestra en el repositorio fuente y decide hacer algunos cambios rápidos para añadir una prueba al código. Li crea un entorno de desarrollo, AWS Cloud9 lo elige como IDE y especifica una nueva rama, *test-code*. Se abre el entorno de desarrollo. Li añade rápidamente el código, luego lo confirma e introduce la rama con los cambios en CodeCatalyst el repositorio fuente. Luego, Li crea una solicitud de extracción. Como parte de la creación de esa solicitud de extracción, Li añade a Jorge Souza y Saanvi Sarkar como revisores para garantizar que se revise el código.

Al revisar el código, Jorge Souza recuerda que tiene su propio repositorio de proyectos que contiene un prototipo de la GitHub aplicación en la que están trabajando. Pide a Mary Major que instale y configure la extensión que le permitirá vincular el GitHub repositorio al proyecto como un repositorio fuente adicional. Mary revisa el repositorio GitHub y trabaja con Jorge para configurar la GitHub extensión para que pueda vincular el GitHub repositorio como un repositorio fuente adicional para el proyecto.

CodeCatalyst Los repositorios de código fuente admiten la funcionalidad estándar de Git y funcionan con las herramientas existentes basadas en Git. Puedes crear y usar tokens de acceso personal (PAT) como contraseña específica de la aplicación al clonar y trabajar con repositorios de código fuente desde un cliente Git o entornos de desarrollo integrados (IDE). Estas PAT están asociadas a tu identidad de usuario. CodeCatalyst Para obtener más información, consulte [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#).

CodeCatalyst Los repositorios de origen admiten solicitudes de cambios. Esta es una forma sencilla para que tú y otros miembros del proyecto revisen y comenten los cambios de código antes de combinarlos de una rama a otra. Puedes ver los cambios en la CodeCatalyst consola y comentar las líneas de código.

Las transferencias a las ramas de un repositorio CodeCatalyst fuente pueden iniciar automáticamente una ejecución en un flujo de trabajo, donde los cambios se pueden crear, probar e implementar. Si el repositorio de origen se creó como parte de un proyecto mediante una plantilla de proyecto, se configurarán uno o varios flujos de trabajo como parte del proyecto. Puedes añadir flujos de trabajo adicionales para los repositorios en cualquier momento. Los archivos de configuración YAML para los flujos de trabajo de un proyecto se almacenan en los repositorios de origen configurados en la acción de origen de esos flujos de trabajo. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).

Temas

- [Conceptos de repositorio de fuentes](#)
- [Configuración para trabajar con repositorios de código fuente](#)
- [Introducción a los repositorios de CodeCatalyst código fuente y al modelo de aplicación de una sola página](#)
- [Trabajar con repositorios de código fuente en CodeCatalyst](#)
- [Trabajar con sucursales en Amazon CodeCatalyst](#)
- [Trabajar con archivos en Amazon CodeCatalyst](#)
- [Trabajar con solicitudes de cambios en Amazon CodeCatalyst](#)
- [Trabajar con compromisos en Amazon CodeCatalyst](#)
- [Cuotas para los repositorios de fuentes en CodeCatalyst](#)

Conceptos de repositorio de fuentes

Estos son algunos conceptos que debe conocer al trabajar con los repositorios de CodeCatalyst código fuente.

Temas

- [Proyectos](#)
- [Repositorios de origen](#)
- [Entornos de desarrollo](#)
- [Tokens de acceso personal \(PAT\)](#)
- [Ramificaciones](#)
- [Ramas predeterminadas](#)
- [Confirmaciones](#)
- [Solicitudes de extracción](#)
- [Revisiones](#)
- [Flujos de trabajo](#)

Proyectos

Un proyecto representa un esfuerzo colaborativo CodeCatalyst que apoya las tareas y los equipos de desarrollo. Una vez que tengas un proyecto, puedes añadir, actualizar o eliminar usuarios y recursos, personalizar el panel de control del proyecto y supervisar el progreso del trabajo de tu equipo. Puede tener varios proyectos dentro de un espacio.

Los repositorios de origen son específicos de los proyectos en los que los creas o enlazas en un espacio. No puedes compartir un repositorio entre proyectos ni puedes vincular un repositorio a más de un proyecto en un espacio. Los usuarios con el rol de colaborador o administrador de proyectos en un proyecto pueden interactuar con los repositorios de origen asociados a ese proyecto de acuerdo con los permisos otorgados a esos roles. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Repositorios de origen

Un repositorio de código fuente es el lugar donde se almacenan de forma segura el código y los archivos del proyecto. También almacena el historial de versiones de los archivos. De forma predeterminada, el repositorio de origen se comparte con los demás usuarios CodeCatalyst del

proyecto. Puedes tener más de un repositorio de código fuente para un proyecto. Puedes crear repositorios de código fuente para proyectos en CodeCatalyst él, o puedes elegir vincular un repositorio de código fuente existente hospedado por otro servicio si ese servicio es compatible con una extensión instalada. Por ejemplo, puedes vincular un GitHub repositorio a un proyecto después de instalar la extensión GitHub Repositories. Para obtener más información, consulte [Trabajar con repositorios de código fuente en CodeCatalyst](#) y [Inicio rápido: Uso de GitHub repositorios en CodeCatalyst](#).

Entornos de desarrollo

Un entorno de desarrollo es un entorno de desarrollo basado en la nube que puede utilizar CodeCatalyst para trabajar rápidamente en el código almacenado en los repositorios de código fuente de su proyecto. Las herramientas del proyecto y las bibliotecas de aplicaciones incluidas en su entorno de desarrollo se definen mediante un archivo de desarrollo en el repositorio de código fuente de su proyecto. Si no tienes un devfile en tu repositorio de código fuente, se aplicará automáticamente un devfile predeterminado. El devfile predeterminado incluye herramientas para los marcos y lenguajes de programación más utilizados. De forma predeterminada, un entorno de desarrollo está configurado para tener un procesador de 2 núcleos, 4 GB de RAM y 16 GiB de almacenamiento persistente.

Puede optar por clonar una rama existente de su repositorio de código fuente en su entorno de desarrollo o puede optar por crear una nueva rama como parte de la creación del entorno de desarrollo.

Tokens de acceso personal (PAT)

Un token de acceso personal (PAT) es similar a una contraseña. Está asociado a tu identidad de usuario para que lo utilices en todos los espacios y proyectos en los que se encuentre CodeCatalyst. Las PATs se utilizan para acceder a CodeCatalyst recursos que incluyen entornos de desarrollo integrados (IDE) y repositorios de código fuente basados en Git. Las PAT te representan CodeCatalyst y puedes administrarlas en tu configuración de usuario. Un usuario puede tener más de una PAT. Los tokens de acceso personal solo se muestran una vez. Como práctica recomendada, asegúrese de guardarlos de forma segura en su ordenador local. De forma predeterminada, los PAT caducan al cabo de un año.

Cuando se trabaja con entornos de desarrollo integrados (IDE), las PAT equivalen a una contraseña de Git. Proporcione la PAT cuando se le solicite una contraseña al configurar su IDE para que funcione con un repositorio de Git. Para obtener más información sobre cómo conectar el IDE con un repositorio basado en Git, consulte la documentación del IDE.

Ramificaciones

Una rama es un puntero o referencia a una confirmación en Git y en CodeCatalyst. Puedes usar ramas para organizar tu trabajo. Por ejemplo, puede usar las ramas para trabajar en una versión nueva o diferente de los archivos sin que ello afecte a los archivos de otras ramas. Puedes usar las ramas para desarrollar nuevas funciones, almacenar una versión específica de tu proyecto y mucho más. Un repositorio de origen puede tener una rama o varias ramas. Al crear un proyecto con una plantilla, el repositorio de origen creado para el proyecto contiene archivos de ejemplo en una rama llamada main. La rama principal es la rama predeterminada del repositorio.

Ramas predeterminadas

Los repositorios de origen CodeCatalyst tienen una rama predeterminada, independientemente de cómo los cree. Si eliges crear un proyecto con una plantilla, el repositorio de origen creado para ese proyecto incluye un archivo README.md además de código de muestra, definiciones de flujo de trabajo y otros recursos. Si creas un repositorio de código fuente sin usar una plantilla, se añade un archivo README.md como primera confirmación y se crea una rama predeterminada como parte de la creación del repositorio. Esta rama predeterminada se denomina main. Esta ramificación predeterminada es la que se utiliza como base o ramificación predeterminada en los repositorios (repos) locales cuando los usuarios clonan el repositorio. Puede cambiar la rama que se usa como rama predeterminada. Para obtener más información, consulte [Ver y cambiar la rama predeterminada de un repositorio](#).

No puedes eliminar la rama predeterminada de un repositorio de origen. Los resultados de la búsqueda solo incluyen los resultados de la rama predeterminada.

Confirmaciones

Una confirmación es un cambio en un archivo o conjunto de archivos. En la CodeCatalyst consola de Amazon, una confirmación guarda los cambios y los envía a un repositorio de origen. La confirmación incluye información sobre el cambio, incluida la identidad del usuario que lo realizó, la fecha y hora del cambio, el título del compromiso y cualquier mensaje incluido sobre el cambio. Para obtener más información, consulte [Trabajar con compromisos en Amazon CodeCatalyst](#).

En el contexto de un repositorio fuente CodeCatalyst, las confirmaciones son instantáneas del contenido del repositorio y de los cambios en él. También puedes añadir etiquetas Git a las confirmaciones para identificar confirmaciones específicas.

Solicitudes de extracción

Una solicitud de incorporación de cambios es la principal forma en que tú y otros usuarios revisan, comentan y fusionan los cambios de código de una rama a otra en un repositorio de código fuente. Puedes usar las solicitudes de extracción para revisar los cambios en el código de forma colaborativa para comprobar si se han realizado cambios o correcciones menores, si se han incorporado funciones importantes o si se han introducido nuevas versiones del software que has publicado. En una solicitud de extracción, puedes revisar los cambios entre las sucursales de origen y destino o las diferencias entre las revisiones de esas sucursales. Puedes añadir comentarios a los cambios de líneas de código individuales, así como comentarios sobre la solicitud de extracción en general.

Tip

Al crear una solicitud de extracción, la diferencia que se muestra es la diferencia entre la punta de la rama de origen y la punta de la rama de destino. Una vez creada la solicitud de extracción, la diferencia que se mostrará será entre la revisión de la solicitud de extracción que hayas elegido y la confirmación que estaba en la punta de la sucursal de destino cuando creaste la solicitud de extracción. Para obtener más información sobre las diferencias y las bases de fusión en Git, consulta [git-merge-base](#) la documentación de Git.

Revisiones

Una revisión es una versión actualizada de una solicitud de cambios. Cada envío a la rama de origen de una solicitud de incorporación de cambios crea una revisión que contiene los cambios realizados en las confirmaciones incluidas en ese envío. Puedes ver las diferencias entre las revisiones de una solicitud de cambios, además de las diferencias entre las ramas de origen y destino. Para obtener más información, consulte [Trabajar con solicitudes de cambios en Amazon CodeCatalyst](#).

Flujos de trabajo

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

i Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Un repositorio de origen también puede almacenar los archivos de configuración y otra información para los flujos de trabajo, las notificaciones, los problemas y otra información de configuración del proyecto. Los archivos de configuración se crean y almacenan en el repositorio de origen cuando se crean recursos que requieren archivos de configuración o cuando se especifica el repositorio como acción de origen para un flujo de trabajo. Si crea un proyecto a partir de un blueprint, ya dispondrá de los archivos de configuración almacenados en el repositorio de origen creado para usted como parte del proyecto. Esta información de configuración se almacena en una carpeta denominada `.codecatalyst` en la rama predeterminada del repositorio. Al crear una rama de la rama predeterminada, se crea una copia de esta carpeta y su configuración, además de todos los demás archivos y carpetas de esa rama.

Configuración para trabajar con repositorios de código fuente

Cuando trabajas con los repositorios de código fuente de Amazon CodeCatalyst en tu máquina local, puedes usar Git solo o en un entorno de desarrollo integrado (IDE) compatible para realizar cambios en el código y enviar y extraer el código. Como práctica recomendada, le recomendamos que use las últimas versiones de Git y otro software.

i Note

Si utilizas Dev Environments, no tienes que instalar Git. Se incluye una versión reciente de Git en tu entorno de desarrollo.

Información de compatibilidad de versiones para CodeCatalyst

Componente	Versión
Git	más recientes

Instalar Git

Para trabajar con archivos, confirmaciones, ramas y otra información de los repositorios de código fuente desde un cliente Git sin un IDE, instala Git en tu máquina local.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Crea un token de acceso personal

Para clonar los repositorios de origen en su máquina local o en su IDE preferido, debe crear un token de acceso personal (PAT).

Para crear un token de acceso personal (PAT)

1. En la barra de menú superior, elija su insignia de perfil y, a continuación, elija Mi configuración.

Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

2. En Nombre de PAT, introduzca un nombre descriptivo para su PAT.
3. En Fecha de caducidad, deje la fecha predeterminada o elija el icono del calendario para seleccionar una fecha personalizada. La fecha de caducidad se establece de forma predeterminada en un año a partir de la fecha actual.
4. Seleccione Crear.

También puede crear este token si elige Clonar repositorio para un repositorio de origen.

5. Guarde el secreto PAT en un lugar seguro.

Important

El secreto del PAT solo se muestra una vez. No podrá recuperarlo después de cerrar la ventana.

Introducción a los repositorios de CodeCatalyst código fuente y al modelo de aplicación de una sola página

Sigue los pasos de este tutorial para aprender a trabajar con los repositorios de código fuente en Amazon CodeCatalyst.

La forma más rápida de empezar a trabajar con los repositorios de código fuente en Amazon CodeCatalyst es crear un proyecto con una plantilla. Cuando creas un proyecto con una plantilla, los recursos se crean automáticamente, incluido un repositorio de código fuente que incluye código de muestra. Puedes usar este repositorio y este ejemplo de código para aprender a:


- Ver los repositorios de fuentes de un proyecto y explorar su contenido
- Crea un entorno de desarrollo con una nueva rama en la que puedas trabajar con el código
- Cambia un archivo y confirma y envía tus cambios
- Crea una solicitud de cambios y revisa los cambios de código con otros miembros del proyecto
- Observa el flujo de trabajo de tu proyecto, crea y prueba automáticamente los cambios en la rama fuente de la solicitud de extracción
- Combina los cambios de tu rama de origen con la de destino y cierra la solicitud de extracción
- Observa cómo se crean e implementan automáticamente los cambios combinados

Para aprovechar al máximo este tutorial, invita a otras personas a tu proyecto para que puedan trabajar juntos en una solicitud de incorporación de cambios. También puedes explorar otras funciones CodeCatalyst, como crear incidencias y asociarlas a una solicitud de extracción, o configurar las notificaciones y recibir alertas cuando se ejecute el flujo de trabajo asociado. Para una exploración completa de CodeCatalyst, consulte [Tutoriales de introducción](#).

Crear un proyecto mediante un plano

Crear un proyecto es el primer paso para poder trabajar juntos. Puedes usar un plano para crear tu proyecto, que también creará un repositorio fuente con código de muestra y un flujo de trabajo que compilará e implementará automáticamente tu código cuando lo cambies. En este tutorial, te mostraremos un proyecto creado con el esquema de aplicación de una sola página, pero puedes seguir los procedimientos de cualquier proyecto con un repositorio de código fuente. Asegúrese de elegir una función de IAM o de añadir una función de IAM si no la tiene como parte de la creación del proyecto. Le recomendamos que utilice el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio para este proyecto.

Si ya tiene un proyecto, puede pasar a [Ver los repositorios de un proyecto](#).

 Note

Solo los usuarios con el rol de administrador de Space o de usuario avanzado pueden crear proyectos en él CodeCatalyst. Si no tienes este rol y necesitas un proyecto en el que trabajar para este tutorial, pídele a alguien con uno de estos roles que cree un proyecto para ti y te añada al proyecto creado. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Para crear un proyecto con un plano

1. En la CodeCatalyst consola, navegue hasta el espacio en el que desee crear un proyecto.
2. En el panel de control del espacio, seleccione Crear proyecto.
3. Seleccione Empezar con un plano.
4. Elija un plano y, a continuación, seleccione Siguiente.
5. En Asigne un nombre al proyecto, introduzca el nombre que desee asignar al proyecto y los nombres de los recursos asociados. El nombre debe ser único dentro de su espacio.
6. En Recursos del proyecto, configure los parámetros comunes del proyecto.
7. (Opcional) Para ver los archivos de definición con actualizaciones basadas en las selecciones de parámetros del proyecto que haya realizado, seleccione Ver código o Ver flujo de trabajo en Generar vista previa del proyecto.
8. (Opcional) Seleccione Ver detalles en la tarjeta del blueprint para ver detalles específicos del blueprint, como una descripción general de la arquitectura del blueprint, las conexiones y los permisos necesarios y el tipo de recursos que crea el blueprint.
9. Elija Crear proyecto.

Para obtener más información sobre los esquemas de los proyectos, consulte [Referencia del plano del proyecto](#)

La página de descripción general del proyecto se abre en cuanto se crea un proyecto o se acepta una invitación para participar en él y se completa el proceso de inicio de sesión. La página de resumen de un proyecto nuevo no contiene temas pendientes ni solicitudes de selección. Si lo desea, puede optar por crear una edición y asignársela a usted mismo. También puedes optar por

invitar a otra persona a tu proyecto. Para obtener más información, consulte [Crear un problema en CodeCatalyst](#) y [Invitar a un usuario a tu proyecto](#).

Ver los repositorios de un proyecto

Como miembro de un proyecto, puedes ver los repositorios de código fuente del proyecto. También puedes optar por crear repositorios adicionales. Si alguien con el rol de administrador de Space ha instalado y configurado la extensión de GitHub repositorios, también puedes añadir enlaces a los GitHub repositorios en GitHub las cuentas configuradas para la extensión. Para obtener más información, consulte [Crear un repositorio de código fuente](#) y [Inicio rápido: Uso de GitHub repositorios en CodeCatalyst](#).

Note

En el caso de los proyectos creados con el esquema de aplicación de una sola página, el nombre predeterminado del repositorio de origen que contiene el código de muestra es spa-app.

Para navegar a los repositorios de origen de un proyecto

1. Navegue hasta su proyecto y realice una de las siguientes acciones:
 - En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio.
 - En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen. En Repositorios de origen, elige el nombre del repositorio de la lista. Puede filtrar la lista de repositorios escribiendo parte del nombre del repositorio en la barra de filtros.
2. En la página de inicio del repositorio, consulta el contenido del repositorio y la información sobre los recursos asociados, como el número de solicitudes de incorporación de datos y los flujos de trabajo. De forma predeterminada, se muestra el contenido de la rama predeterminada. Puede cambiar la vista seleccionando una rama diferente de la lista desplegable.

La página de información general del repositorio incluye información sobre los flujos de trabajo y las solicitudes de extracción configurados para las ramas de este repositorio y sus archivos. Si acabas de crear el proyecto, los flujos de trabajo iniciales para crear, probar e implementar el código seguirán ejecutándose, ya que tardan unos minutos en completarse. Puede ver los flujos de trabajo relacionados y su estado seleccionando el número que aparece debajo de Flujos de trabajo

relacionados, pero esto abrirá la página Flujos de trabajo en CI/CD. Para ver este tutorial, visita la página de información general y explora el código del repositorio. El contenido del README .md archivo se muestra en esta página, debajo de los archivos del repositorio. En Archivos, se muestra el contenido de la rama predeterminada. Puede cambiar la vista del archivo para mostrar el contenido de otra rama, si tiene una. La `.codecatalyst` carpeta contiene el código utilizado para otras partes del proyecto, como los archivos YAML del flujo de trabajo.

Para ver el contenido de las carpetas, elija la flecha situada junto al nombre de la carpeta para ampliarla. Por ejemplo, elija la flecha situada junto a esta `src` para ver los archivos de la aplicación web de una sola página contenida en esa carpeta. Para ver el contenido de un archivo, elíjalo en la lista. Se abrirá la opción Ver archivos, donde podrá examinar el contenido de varios archivos. También puedes editar archivos individuales en la consola, pero para editar varios archivos, querrás crear un entorno de desarrollo.

Creación de un entorno de desarrollo

Puedes añadir y cambiar archivos en un repositorio de origen de la CodeCatalyst consola de Amazon. Sin embargo, para trabajar eficazmente con varios archivos y ramas, te recomendamos utilizar un entorno de desarrollo o clonar el repositorio en tu ordenador local. En este tutorial, crearemos un entorno de AWS Cloud9 desarrollo con una rama denominada **develop**. Puedes elegir un nombre de rama diferente, pero si le pones un nombre a la rama **develop**, se ejecutará automáticamente un flujo de trabajo para compilar y probar tu código cuando crees una solicitud de extracción más adelante en este tutorial.

Tip

Si decides clonar un repositorio de forma local en lugar de utilizar un entorno de desarrollo o además de él, asegúrate de tener Git en tu ordenador local o de que tu IDE incluye Git. Para obtener más información, consulte [Configuración para trabajar con repositorios de código fuente](#).

Para crear un entorno de desarrollo con una nueva rama

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya al proyecto donde desea crear un entorno de desarrollo.

3. Elija el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código, elige Repositorios de código fuente y elige el repositorio para el que quieres crear un entorno de desarrollo.
4. En la página de inicio del repositorio, selecciona Crear entorno de desarrollo.
5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Elija el repositorio que desea clonar, elija Trabajar en una nueva ramificación, introduzca el nombre de la ramificación en el campo Nombre de ramificación, y elija una ramificación a partir de la cual crear la ramificación nueva en el menú desplegable Crear ramificación desde.
7. Si lo desea, añada un alias para el entorno de desarrollo.
8. Si lo desea, elija el botón de editar Configuración del entorno de desarrollo para editar la configuración informática, de almacenamiento o el tiempo de espera del entorno de desarrollo.
9. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo. Se abrirá una nueva pestaña con su entorno de desarrollo en el IDE que elija. Puedes editar el código y confirmar y enviar los cambios.

Una vez que hayas creado el entorno de desarrollo, puedes editar los archivos, confirmar los cambios y enviarlos a la **test** rama. En este tutorial, edita el contenido entre las `<p>` etiquetas del `App.tsx` archivo de la `src` carpeta para cambiar el texto que se muestra en la página web. Confirme y presione el cambio y, a continuación, regrese a la CodeCatalyst pestaña.

Para realizar e impulsar un cambio desde un entorno AWS Cloud9 de desarrollo

1. En AWS Cloud9, expande el menú de navegación lateral para buscar los archivos. `src`Expande y abre `App.tsx`.
2. Realice un cambio en el texto dentro de las `<p>` etiquetas.
3. Guarde el archivo y, a continuación, confirme e inserte los cambios mediante el menú Git. Como alternativa, en la ventana del terminal, confirma y envía los cambios con los `git push` comandos `git commit` y.

```
git commit -am "Making an example change"
git push
```

 Tip

Es posible que tengas que cambiar los directorios de la terminal al directorio del repositorio de Git para poder ejecutar correctamente los comandos de Git.

Creación de una solicitud de extracción

Puedes utilizar las solicitudes de extracción para revisar los cambios en el código de forma colaborativa para comprobar si se han realizado cambios o correcciones menores, si se han incorporado funciones importantes o si se han introducido nuevas versiones del software que hayas publicado. En este tutorial, crearás una solicitud de extracción para revisar los cambios que has realizado en la rama de *prueba* en comparación con la rama principal. Al crear una solicitud de extracción en un proyecto creado con una plantilla, también se iniciarán los flujos de trabajo asociados, si los hubiera.

Para crear una solicitud de extracción


1. Vaya a su proyecto.
2. Realice una de las acciones siguientes:
 - En el panel de navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, selecciona Crear solicitud de extracción.
 - En la página de inicio del repositorio, selecciona Más y, a continuación, selecciona Crear solicitud de extracción.
 - En la página del proyecto, selecciona Crear solicitud de extracción.
3. En el repositorio de fuentes, asegúrate de que el repositorio de fuentes especificado es el que contiene el código confirmado. Esta opción solo aparece si no creaste la solicitud de extracción desde la página principal del repositorio.
4. En la rama de destino, elige la rama en la que deseas combinar el código después de revisarlo.
5. En la rama Fuente, elige la rama que contiene el código confirmado.
6. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios a entender qué es lo que hay que revisar y por qué.
7. (Opcional) En la descripción de la solicitud de extracción, proporciona información como un enlace a los problemas o una descripción de tus cambios.

 Tip

Puedes elegir Escribir una descripción para que yo genere CodeCatalyst automáticamente una descripción de los cambios incluidos en la solicitud de cambios. Puedes realizar cambios en la descripción generada automáticamente después de añadirla a la solicitud de extracción.


Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

8. (Opcional) En Problemas, selecciona Vincular problemas y, a continuación, elige un problema de la lista o introduce su ID. Para desvincular una incidencia, selecciona el icono de desvinculación.
9. (Opcional) En Revisores obligatorios, selecciona Agregar revisores obligatorios. Elija de la lista de miembros del proyecto para añadirlos. Los revisores necesarios deben aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.

 Note

No puedes añadir un revisor como revisor obligatorio y como revisor opcional. No puedes añadirte a ti mismo como revisor.

10. (Opcional) En Revisores opcionales, selecciona Añadir revisores opcionales. Elija de la lista de miembros del proyecto para añadirlos. Los revisores opcionales no tienen que aprobar los cambios como requisito para poder fusionar la solicitud de extracción en la sucursal de destino.
11. Revisa las diferencias entre las sucursales. La diferencia que se muestra en una solicitud de extracción son los cambios entre la revisión en la rama de origen y la base de fusión, que es la confirmación principal de la rama de destino en el momento en que se creó la solicitud de extracción. Si no se muestra ningún cambio, es posible que las ramas sean idénticas o que hayas elegido la misma rama tanto para el origen como para el destino.
12. Cuando estés seguro de que la solicitud de extracción contiene el código y los cambios que deseas revisar, selecciona Crear.

 Note

Después de crear la solicitud de extracción, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de

los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a recursos, como archivos, utilizando el signo @ seguido del nombre del archivo.

Para ver la información sobre los flujos de trabajo asociados iniciados con la creación de esta solicitud de extracción, selecciona Descripción general y, a continuación, revisa la información en el área de detalles de la solicitud de extracción, en Ejecuciones del flujo de trabajo. Para ver la ejecución del flujo de trabajo, selecciónela.

Tip

Si has asignado a la rama un nombre diferente al de «**develop**, no se ejecutará automáticamente un flujo de trabajo para crear y probar los cambios». Si quieres configurarlo, edita el archivo YAML del onPullRequestBuildAndTestflujo de trabajo. Para obtener más información, consulte [Crear, editar y eliminar un flujo de trabajo](#).

Puedes comentar esta solicitud de cambios y pedir a otros miembros del proyecto que comenten sobre ella. También puedes optar por añadir o cambiar revisores opcionales u obligatorios. Puedes optar por realizar más cambios en la rama de origen del repositorio y ver cómo esos cambios confirmados crean revisiones para la solicitud de extracción de información. Para obtener más información [Revisión de una solicitud de extracción](#), consulte [Actualización de una solicitud de extracción](#), [Trabajar con solicitudes de cambios en Amazon CodeCatalyst](#), y [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Fusionar una solicitud de extracción

Una vez que se haya revisado una solicitud de extracción y haya recibido la aprobación de los revisores necesarios, puedes fusionar su rama de origen con la de destino en la CodeCatalyst consola. Al fusionar una solicitud de extracción, también se iniciarán los cambios en todos los flujos de trabajo asociados a la sucursal de destino. En este tutorial, fusionarás la rama de prueba con la principal, lo que iniciará una ejecución del onPushToMainDeployPipelineflujo de trabajo.

Para fusionar una solicitud de extracción (consola)

1. En las solicitudes de extracción, elige la solicitud de extracción que creaste en el paso anterior. En la solicitud de extracción, selecciona Fusionar.

2. Elige entre las estrategias de fusión disponibles para la solicitud de extracción. Si lo deseas, selecciona o deselecciona la opción para eliminar la rama de origen después de fusionar la solicitud de extracción y, a continuación, selecciona Fusionar. Una vez completada la fusión, el estado de la solicitud de extracción cambia a Combinada y deja de aparecer en la vista predeterminada de las solicitudes de extracción. La vista predeterminada muestra las solicitudes de extracción con el estado Abierto. Aún puedes ver una solicitud de extracción combinada, pero no puedes aprobarla ni cambiar su estado.

Note

Si el botón Combinar no está activo o ves la etiqueta No se puede combinar, significa que un revisor obligatorio aún no ha aprobado la solicitud de extracción o la solicitud de extracción no se puede fusionar en la CodeCatalyst consola. Un revisor que no ha aprobado una solicitud de extracción se indica con un icono de reloj en la sección Descripción general, en el área de detalles de la solicitud de extracción. Si todos los revisores necesarios han aprobado la solicitud de extracción, pero el botón Combinar sigue sin estar activo, es posible que tengas un conflicto de fusión o que hayas superado la cuota de almacenamiento del espacio. Puedes resolver los conflictos de fusión para la rama de destino en un entorno de desarrollo, insertar los cambios y, a continuación, fusionar la solicitud de extracción, o puedes resolver los conflictos y fusionar localmente y, a CodeCatalyst continuación, presionar la confirmación que contiene la fusión. Para obtener más información, consulta [Fusionar una solicitud de cambios \(Git\)](#) la documentación de Git.

Visualización del código desplegado

Ahora es el momento de ver el código implementado originalmente que estaba en la rama predeterminada y los cambios combinados una vez que se hayan creado, probado e implementado automáticamente. Para ello, puede volver a la página de información general del repositorio y elegir el número situado junto al icono de los flujos de trabajo relacionados o, en el panel de navegación, elegir CI/CD y, a continuación, elegir Flujos de trabajo.

Para ver el código desplegado

1. En Flujos de trabajo, en `nonPushToMainDeployPipeline`, expanda Ejecuciones recientes.

Note

Este es el nombre predeterminado del flujo de trabajo para los proyectos creados con el blueprint de la aplicación de una sola página.

2. La ejecución más reciente es la que se inició con la confirmación de la solicitud de cambios fusionada con la main rama y es probable que muestre el estado En curso. Selecciona una ejecución completada correctamente de la lista para abrir los detalles de esa ejecución.
3. Elija Variables. Copie el valor de AppURL. Esta es la URL de la aplicación web de una sola página implementada. Abra una nueva pestaña del navegador y pegue el valor para ver el código creado e implementado. Deje la pestaña abierta.
4. Vuelva a la lista de ejecuciones del flujo de trabajo y espere a que se complete la ejecución más reciente. Cuando lo haga, regrese a la pestaña que abrió para ver la aplicación web y actualizar el navegador. Deberás ver los cambios que has realizado en tu solicitud de extracción combinada.

Limpieza de recursos

Una vez que hayas pensado en trabajar con un repositorio de código fuente y una solicitud de extracción, es posible que desees eliminar los recursos que no necesites. No puedes eliminar las solicitudes de extracción, pero puedes cerrarlas. Puedes eliminar cualquier rama que hayas creado.

Si ya no necesitas el repositorio de origen o el proyecto, también puedes eliminar esos recursos. Para obtener más información, consulte [Eliminar un repositorio de fuentes](#) y [Eliminar un proyecto en Amazon CodeCatalyst](#).

Trabajar con repositorios de código fuente en CodeCatalyst

Un repositorio de código fuente es el lugar donde se almacenan de forma segura el código y los archivos de su proyecto. También almacena el historial de fuentes, desde la primera confirmación hasta los cambios más recientes. Si eliges un plano que incluya un repositorio de origen, ese repositorio también incluirá los archivos de configuración y otra información para los flujos de trabajo y las notificaciones del proyecto. Esta información de configuración se almacena en una carpeta denominada .codecatalyst.

Puede crear un repositorio de origen creando un proyecto con un esquema que cree un repositorio de origen como parte de la creación de un proyecto, o CodeCatalyst bien creando un repositorio de código fuente en un proyecto existente. Los usuarios del proyecto verán automáticamente los repositorios que crees para un proyecto y podrán usarlos. También puedes optar por vincular un repositorio de Git alojado en GitHub tu proyecto. Cuando lo hagas, los usuarios del proyecto podrán ver y acceder a ese repositorio vinculado en la lista de repositorios del proyecto.

Note

Para poder vincular el repositorio, debes instalar la extensión para el servicio que lo aloja. No puede vincular un repositorio archivado. Si bien puedes vincular un repositorio vacío, no puedes usarlo CodeCatalyst hasta que lo hayas inicializado con una confirmación inicial que cree una rama predeterminada. Para obtener más información, consulte [Instalación de una extensión](#).

De forma predeterminada, el repositorio de origen se comparte con otros miembros de tu CodeCatalyst proyecto de Amazon. Puedes crear repositorios de fuentes adicionales para un proyecto o vincular repositorios al proyecto. Todos los miembros de un proyecto pueden ver, añadir, editar y eliminar archivos y carpetas en los repositorios de origen del proyecto.

Para trabajar rápidamente en el código de un repositorio de código fuente, puedes crear un entorno de desarrollo que clone un repositorio específico y ramificarlo en él para poder trabajar con el código en el entorno de desarrollo integrado (IDE) que elijas para el entorno de desarrollo. Puedes clonar un repositorio de código fuente en tu ordenador local y extraer e insertar los cambios entre tu repositorio local y el repositorio remoto. CodeCatalyst También puedes trabajar con los repositorios de código fuente configurando el acceso a ellos en el IDE que prefieras, siempre que dicho IDE permita la administración de credenciales.

Los nombres de los repositorios deben ser únicos dentro de un proyecto CodeCatalyst .

Temas

- [Crear un repositorio de código fuente](#)
- [Vincular un repositorio de fuentes](#)
- [Visualización de un repositorio de fuentes](#)
- [Edición de la configuración de un repositorio de fuentes](#)
- [Clonación de un repositorio fuente](#)

- [Eliminar un repositorio de fuentes](#)

Crear un repositorio de código fuente

Cuando creas un proyecto con un blueprint en Amazon CodeCatalyst, CodeCatalyst crea un repositorio de código fuente para ti. Ese repositorio fuente contiene código de muestra además de información de configuración para los flujos de trabajo y otros recursos creados para usted. Esta es la forma recomendada de empezar con los repositorios en CodeCatalyst. Puede optar por crear repositorios para un proyecto. Estos repositorios contendrán un único archivo, un archivo README.md que podrás editar o eliminar en cualquier momento. En función de las opciones que elija al crear un repositorio de origen, es posible que los repositorios también contengan un archivo `.gitignore`.

Para crear un repositorio de origen

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su proyecto.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
5. En Nombre del repositorio, especifique un nombre para el repositorio. En esta guía, utilizamos *codecatalyst-source-repository*, pero puede elegir un nombre diferente. Los nombres de repositorio deben ser únicos dentro de un proyecto. Para obtener más información sobre los requisitos de los nombres de los repositorios, consulte [Cuotas para los repositorios de fuentes en CodeCatalyst](#).
6. (Opcional) En la descripción, añada una descripción del repositorio que ayude a los demás usuarios del proyecto a entender para qué se utiliza el repositorio.
7. (Opcional) Agrega un `.gitignore` archivo para el tipo de código que planeas insertar.
8. Seleccione Crear.

Note

CodeCatalyst agrega un README .md archivo a tu repositorio al crearlo. CodeCatalyst también crea una confirmación inicial para el repositorio en una rama predeterminada llamada main. Puede editar o eliminar el archivo README.md, pero no puede cambiar ni eliminar la rama predeterminada.

Vincular un repositorio de fuentes

Al vincular un repositorio de origen a un proyecto, puedes incluir repositorios que tengan una CodeCatalyst extensión para el servicio que aloja el repositorio, si esa extensión está instalada en tu espacio. Solo los usuarios con el rol de administrador de Space pueden instalar extensiones. Una vez instalada la extensión, puede establecer un enlace a los repositorios configurados para el acceso mediante esa extensión. Para obtener más información, consulte [Instalación y desinstalación de extensiones en CodeCatalyst](#) o siga [Inicio rápido: Uso de GitHub repositorios en CodeCatalyst](#).

Note

Puede vincular un repositorio a un solo proyecto en un espacio. No puede vincular un repositorio archivado. Si bien puedes vincular un repositorio vacío, no puedes usarlo CodeCatalyst hasta que lo hayas inicializado con una confirmación inicial que cree una rama predeterminada.

Para vincular un repositorio de origen

1. Navegue hasta el proyecto al que desee vincular un repositorio.

Note

Para poder vincular un repositorio, un usuario con el rol de administrador de Space debe instalar primero la extensión para el proveedor que aloja el repositorio. Para obtener más información, consulte [Instalación de una extensión](#).

2. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
3. Selecciona Añadir repositorio y, a continuación, selecciona Vincular repositorio.
4. En Proveedor de repositorios, elija el nombre del proveedor. En GitHub Cuenta, elige la cuenta vinculada que quieres usar para trabajar en el repositorio y, a continuación, en Repositorio, elige el nombre del repositorio.

Note

Solo puedes vincular un repositorio activo. No puede vincular un repositorio archivado.

5. Revise la información y, a continuación, seleccione Enlazar.

Note

Solo puede vincular un repositorio a un proyecto de un espacio.

Para desvincular un repositorio de un proyecto, debe tener el rol de administrador del espacio. Para obtener más información, consulte [Administrar GitHub repositorios en CodeCatalyst](#).

Visualización de un repositorio de fuentes

Puedes ver los repositorios de origen asociados a un proyecto en Amazon CodeCatalyst. En el caso de los repositorios de origen CodeCatalyst, la página de información general de un repositorio proporciona un resumen rápido de la información y la actividad de ese repositorio, que incluye:

- La descripción del repositorio, si existe
- El número de sucursales del repositorio
- El número de solicitudes de cambios abiertas para el repositorio
- El número de flujos de trabajo relacionados con el repositorio
- Los archivos y carpetas de la rama predeterminada o de la rama que elija
- El título, el autor y la fecha de la última confirmación en la rama mostrada
- El contenido del archivo README.md renderizado en Markdown, si se incluye algún archivo README.md

Esta página también proporciona enlaces a las confirmaciones, ramificaciones y solicitudes de extracción del repositorio, así como una forma rápida de abrir, ver y editar archivos individuales.

Note

No puedes ver esta información sobre los repositorios enlazados en la CodeCatalyst consola. Para ver información sobre los repositorios enlazados, elija el enlace de la lista de repositorios para abrir ese repositorio en el servicio que lo aloja.

Para navegar a los repositorios de origen de un proyecto

1. Navegue hasta su proyecto y realice una de las siguientes acciones:

- En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio.
 - En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen. En Repositorios de origen, elige el nombre del repositorio de la lista. Puede filtrar la lista de repositorios escribiendo parte del nombre del repositorio en la barra de filtros.
2. En la página de inicio del repositorio, consulta el contenido del repositorio y la información sobre los recursos asociados, como el número de solicitudes de incorporación de datos y los flujos de trabajo. De forma predeterminada, se muestra el contenido de la rama predeterminada. Puede cambiar la vista seleccionando una rama diferente de la lista desplegable.

Tip

También puedes navegar rápidamente a los repositorios de tu proyecto seleccionando Ver el código del proyecto en la página de resumen del proyecto.

Edición de la configuración de un repositorio de fuentes

Puedes gestionar la configuración de tu repositorio, incluida la edición de la descripción de un repositorio, la elección de la rama predeterminada, la creación y gestión de las reglas de sucursal y la creación y gestión de las reglas de aprobación para las solicitudes de incorporación de datos CodeCatalyst. Esto puede ayudar a los miembros del proyecto a entender para qué se utiliza el repositorio y ayudarte a aplicar las mejores prácticas y los procesos utilizados por el equipo.

Note

No puedes editar el nombre de un repositorio de origen.
No puedes editar el nombre, la descripción u otra información de un repositorio vinculado en CodeCatalyst. Para modificar la información sobre un repositorio vinculado, debe editarla en el proveedor que aloja el repositorio vinculado. Para obtener más información, consulte la documentación del servicio que aloja el repositorio vinculado.

Para editar la configuración de un repositorio

1. En la CodeCatalyst consola, navegue hasta el proyecto que contiene el repositorio de origen cuya configuración desee editar.

2. En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen. Elija el nombre del repositorio en la lista de repositorios de origen del proyecto.
3. En la página de información general del repositorio, selecciona Más y, a continuación, selecciona Administrar configuración.
4. Realice una o más de las siguientes acciones:
 - Edita la descripción del repositorio y, a continuación, selecciona Guardar.
 - Para cambiar la rama predeterminada del repositorio, en Rama predeterminada, selecciona Editar. Para obtener más información, consulte [Ver y cambiar la rama predeterminada de un repositorio](#).
 - Para añadir, eliminar o cambiar una regla según la cual los roles del proyecto tienen permiso para realizar determinadas acciones en una rama, en Reglas de sucursal, selecciona Editar. Para obtener más información, consulte [Gestione las acciones permitidas para una sucursal con las reglas de la sucursal](#).
 - Para añadir, eliminar o cambiar una regla de aprobación para fusionar solicitudes de extracción con una rama, en Reglas de aprobación, selecciona Editar. Para obtener más información, consulte [Gestione los requisitos para fusionar una solicitud de extracción con las reglas de aprobación](#).

Clonación de un repositorio fuente

Para trabajar eficazmente con varios archivos, ramas y confirmaciones en los repositorios de código fuente, clona el repositorio de código fuente en tu ordenador local y utiliza un cliente Git o un entorno de desarrollo integrado (IDE) para realizar cambios. Confirma los cambios e introdúcelos en el repositorio de código fuente para trabajar con CodeCatalyst funciones como las incidencias y las solicitudes de incorporación de cambios. También puedes optar por crear un entorno de desarrollo para trabajar con el código. Al crear un entorno de desarrollo, se clona automáticamente el repositorio y la rama que especifiques en el entorno de desarrollo.

Note

No puede clonar repositorios enlazados en la CodeCatalyst consola ni crear entornos de desarrollo para ellos. Para clonar un repositorio vinculado de forma local, selecciona el enlace de la lista de repositorios para abrir ese repositorio en el servicio que lo aloja y, a

continuación, clonarlo. Para obtener más información, consulte la documentación del servicio que aloja el repositorio vinculado.

Para crear un entorno de desarrollo a partir de un repositorio fuente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
3. Elige el repositorio de código fuente en el que quieres trabajar con el código.
4. Seleccione Crear entorno de desarrollo.
5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Realice una de las acciones siguientes:
 - Seleccione Trabajar en una rama existente y, a continuación, seleccione una rama en el menú desplegable de la rama existente.
 - Seleccione Trabajar en una nueva sucursal, ingresa el nombre de una sucursal en el campo Nombre de la sucursal y elige una sucursal a partir de la cual crear la nueva rama en el menú desplegable Crear rama desde.
7. Si lo desea, añada un nombre para el entorno de desarrollo o edite su configuración.
8. Seleccione Crear.

Para clonar un repositorio fuente

1. Vaya a su proyecto.
2. En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio. Como alternativa, en el panel de navegación, seleccione Código y, a continuación, seleccione Repositorios de origen. Elija el nombre del repositorio en la lista de repositorios de origen del proyecto. Puede filtrar la lista de repositorios escribiendo parte del nombre del repositorio en la barra de filtros.
- 3.
4. Seleccione Clonar repositorio. Copia la URL del clon del repositorio.

Note

Si no tiene un token de acceso personal (PAT), elija Crear token. Copie el token y guárdelo en un lugar seguro. Utilizará esta PAT cuando su cliente Git o su entorno de desarrollo integrado (IDE) le pida una contraseña.

5. Realice una de las acciones siguientes:

- Para clonar un repositorio en su ordenador local, abra un terminal o una línea de comandos y ejecute el git clone comando con la URL del clon después del comando. Por ejemplo:

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Cuando se le pida una contraseña, pegue la PAT que guardó anteriormente.

Note

Si su sistema operativo proporciona administración de credenciales o ha instalado un sistema de administración de credenciales, solo necesita proporcionar la PAT una vez. De lo contrario, es posible que tengas que proporcionar la PAT para cada operación de Git. Como práctica recomendada, asegúrate de que tu sistema de administración de credenciales almacene tu PAT de forma segura. No incluya la PAT como parte de la cadena URL del clon.

- Para clonar un repositorio mediante un IDE, siga la documentación del IDE. Elige la opción de clonar un repositorio de Git y proporciona la URL. Cuando se le pida una contraseña, proporcione la PAT.

Eliminar un repositorio de fuentes

Si un repositorio de código fuente para un CodeCatalyst proyecto de Amazon ya no es necesario, puedes eliminarlo. Al eliminar un repositorio de origen, también se elimina cualquier información del proyecto almacenada en el repositorio. Si algún flujo de trabajo depende del repositorio de origen, dichos flujos de trabajo se eliminarán de la lista de flujos de trabajo del proyecto después de eliminar el repositorio. Los problemas que hagan referencia al repositorio de origen no se eliminarán ni

modificarán, pero cualquier enlace al repositorio de origen que se añada a los problemas fallará una vez que se elimine el repositorio.

Important

La eliminación de un repositorio fuente no se puede deshacer. Tras eliminar un repositorio de origen, ya no podrá clonarlo, extraer datos de él ni enviarlos a él. Al eliminar un repositorio de origen, no se elimina ninguna copia local de ese repositorio (repositorios locales). Para eliminar un repositorio local, usa las herramientas de administración de archivos y directorios de tu computadora local.

Note

No puede eliminar un repositorio vinculado en la CodeCatalyst consola. Para eliminar un repositorio vinculado, elija el enlace de la lista de repositorios para abrirlo en el servicio que lo aloja y, a continuación, elimínelo. Para obtener más información, consulte la documentación del servicio que aloja el repositorio vinculado.

Para eliminar un repositorio vinculado de un proyecto, consulte [Administrar GitHub repositorios en CodeCatalyst](#).

Para eliminar un repositorio de origen

1. Navegue hasta el proyecto que contiene el repositorio de origen que desea eliminar.
2. En la página de resumen de su proyecto, elija el repositorio que desee de la lista y, a continuación, elija Ver repositorio. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen. Elija el nombre del repositorio en la lista de repositorios de origen del proyecto.
3. En la página de inicio del repositorio, selecciona Más y, a continuación, selecciona Eliminar repositorio.
4. Revisa la rama, la solicitud de extracción y la información relacionada con el flujo de trabajo para asegurarte de no eliminar un repositorio que aún esté en uso o que tenga trabajos pendientes. Si quieres continuar, escribe eliminar y, a continuación, selecciona Eliminar.

Trabajar con sucursales en Amazon CodeCatalyst

En Git, las ramificaciones son indicadores o referencias a una confirmación. Durante el desarrollo, son una cómoda forma de organizar el trabajo. Puede utilizar las ramificaciones para separar el trabajo en una versión de los archivos nueva o diferente sin que ello afecte al trabajo en otras ramificaciones. Puedes usar las sucursales para desarrollar nuevas funciones, almacenar una versión específica de tu proyecto y mucho más. Puedes configurar reglas para las ramas en los repositorios de origen a fin de limitar determinadas acciones de una rama a funciones específicas de ese proyecto.

Los repositorios de fuentes de Amazon CodeCatalyst tienen contenido y una rama predeterminada, independientemente de cómo los crees. Es posible que los repositorios enlazados no tengan una rama o contenido predeterminados, pero no se podrán utilizar CodeCatalyst hasta que los inicialices y crees una rama predeterminada. Al crear un proyecto mediante un plano, CodeCatalyst crea un repositorio de código fuente para ese proyecto que incluye un archivo README.md, un código de muestra, definiciones de flujos de trabajo y otros recursos. Al crear un repositorio de código fuente sin utilizar un plano, se añade un archivo README.md como primera confirmación y se crea una rama predeterminada. Esta rama predeterminada se denomina main. Esta ramificación predeterminada es la que se utiliza como base o ramificación predeterminada en los repositorios (repos) locales cuando los usuarios clonan el repositorio.

Note

No puedes eliminar la rama predeterminada. La primera rama que se crea para un repositorio de origen es la rama predeterminada de ese repositorio. Además, la búsqueda solo muestra los resultados de la rama predeterminada. No puedes buscar código en otras sucursales.

Al crear un repositorio en, CodeCatalyst también se crea una primera confirmación, que crea una rama predeterminada con un archivo README.md incluido. El nombre de esa rama predeterminada es main. Esta es la ramificación predeterminada en todos los ejemplos de esta guía.

Temas

- [Crear y eliminar una rama](#)
- [Ver y cambiar la rama predeterminada de un repositorio](#)
- [Gestione las acciones permitidas para una sucursal con las reglas de la sucursal](#)
- [Comandos de Git para sucursales](#)

- [Visualización de sucursales y detalles](#)

Crear y eliminar una rama

Puedes usar la CodeCatalyst consola para crear y eliminar ramas en un CodeCatalyst repositorio. Las ramas que crees estarán visibles para otros usuarios la próxima vez que extraigan cambios del repositorio. Al eliminar una rama, las copias de esa rama permanecen en los clones del repositorio en los ordenadores locales hasta que los usuarios extraigan esos cambios y los sincronicen.

Tip

También puedes crear ramas como parte de la creación de un entorno de desarrollo para trabajar en tu código. Para obtener más información, consulte [Creación de un entorno de desarrollo](#).

También puedes usar Git para crear y eliminar ramas. Para obtener más información, consulte [Comandos comunes de Git para sucursales](#).

Para crear una rama (consola)

1. En la CodeCatalyst consola, navega hasta el proyecto en el que reside tu repositorio de origen.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.
3. Elija el repositorio en el que desee crear una rama.
4. En la página de información general del repositorio, selecciona Más y, a continuación, selecciona Crear rama.
5. Introduzca un nombre para la rama.
6. Elige una rama desde la que crear la rama y, a continuación, selecciona Crear.

Para eliminar una rama (consola)

1. Navegue hasta el proyecto en el que reside su repositorio.

2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee eliminar una rama.

3. En la página de información general del repositorio, selecciona el selector desplegable situado junto al nombre de la rama y, a continuación, selecciona Ver todo.
4. Elige la rama que quieres eliminar y, a continuación, selecciona Eliminar rama.

Note

No puedes eliminar la rama predeterminada de un repositorio.

5. Aparece un cuadro de diálogo de confirmación. Muestra el repositorio, el número de solicitudes de extracción abiertas y el número de flujos de trabajo asociados a la sucursal.
6. Para confirmar la eliminación de la rama, escribe eliminar en el cuadro de texto y, a continuación, selecciona Eliminar.

Ver y cambiar la rama predeterminada de un repositorio

Puedes especificar qué rama usar como rama predeterminada en un repositorio de origen en Amazon CodeCatalyst. Todos los repositorios de origen CodeCatalyst tienen contenido y una rama predeterminada, independientemente de cómo los crees. Si utilizas un plano para crear un proyecto, la rama predeterminada del repositorio de origen creado para ese proyecto se denomina main. El contenido de la rama predeterminada se muestra automáticamente en la página de información general de ese repositorio.

La rama predeterminada se trata de forma un poco diferente a todas las demás ramas de un repositorio de origen. Tiene una etiqueta especial junto a su nombre, Default. La rama predeterminada es la que se usa como base o rama predeterminada en los repositorios locales (repositorios) cuando los usuarios clonan el repositorio en ordenadores locales con un cliente Git. También es la que se usa por defecto al crear flujos de trabajo para almacenar los archivos YAML del flujo de trabajo y para almacenar información relacionada con los problemas. Al utilizar la búsqueda en CodeCatalyst, solo se busca en la rama predeterminada de un repositorio. Como la rama predeterminada es fundamental para muchos aspectos de los proyectos, no se puede eliminar una rama si está especificada como la rama predeterminada. Sin embargo, puede optar por utilizar una rama diferente como rama predeterminada. Si lo hace, cualquier [regla de sucursal](#) que se

haya aplicado a la antigua sucursal predeterminada se aplicará automáticamente a la sucursal que especifique como sucursal predeterminada.

 Note


Debe tener el rol de administrador de proyectos para cambiar la rama predeterminada de los repositorios de origen en los CodeCatalyst proyectos. Esto no se aplica a los repositorios enlazados.

Para ver y cambiar la rama predeterminada de un repositorio

1. Navegue hasta el proyecto en el que reside su repositorio.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee ver la configuración, incluida la rama predeterminada.

3. En la página de información general del repositorio, selecciona Más y, a continuación, selecciona Administrar configuración.
4. En la rama predeterminada, se muestra el nombre de la rama especificada como la rama predeterminada junto con una etiqueta llamada Predeterminada junto al nombre. Esta misma etiqueta aparece junto al nombre de la sucursal en la lista de sucursales.
5. Para cambiar la rama predeterminada, selecciona Editar.

 Note

Debe tener el rol de administrador del proyecto en el proyecto para cambiar la rama predeterminada.

6. Elija el nombre de la rama que desee convertir en la rama predeterminada en la lista desplegable y, a continuación, seleccione Guardar.

Gestione las acciones permitidas para una sucursal con las reglas de la sucursal

Al crear una rama, se permiten ciertas acciones para esa rama en función de los permisos de esa función. Puedes cambiar las acciones que se permiten en una rama específica configurando las reglas de la sucursal. Las reglas de sucursal se basan en la función que desempeña un usuario en el proyecto. Puedes optar por limitar algunas acciones predefinidas, como enviar las confirmaciones a una sucursal, a los usuarios con un rol concreto en un proyecto. Esto puede ayudarte a proteger ramas específicas de un proyecto al limitar los roles que pueden realizar determinadas acciones. Por ejemplo, si configuras una regla de ramificación para que solo los usuarios con el rol de administrador del proyecto puedan fusionarse o transferirse a esa rama, los usuarios con otros roles en el proyecto no podrán realizar cambios en el código de esa rama.

Deberías considerar detenidamente todas las implicaciones de crear una regla para una rama. Por ejemplo, si decides limitar las entradas a una rama a los usuarios con la función de administrador del proyecto, los usuarios con la función de colaborador no podrán crear ni editar flujos de trabajo en esa rama, ya que el flujo de trabajo YAML se almacena en esa rama y esos usuarios no pueden confirmar ni enviar los cambios al YAML. Como práctica recomendada, prueba las reglas de las sucursales después de crearlas para asegurarte de que no tengan ningún impacto que no esperabas. También puedes usar las reglas de sucursal junto con las reglas de aprobación para las solicitudes de atracción. Para obtener más información, consulte [Gestione los requisitos para fusionar una solicitud de extracción con las reglas de aprobación](#).

Note

Debes tener la función de administrador de proyectos para gestionar las reglas de ramificación de los repositorios de origen de los CodeCatalyst proyectos. No puedes crear reglas de ramificación para los repositorios enlazados.

Solo puede crear reglas de ramificación que sean más restrictivas que los permisos predeterminados para el rol. No puede crear reglas de ramificación que sean más permisivas de lo que permite el rol de un usuario en el proyecto. Por ejemplo, no puedes crear una regla de ramificación que permita a los usuarios con el rol de revisor acceder a la rama.

Las reglas de bifurcación que se aplican a la rama predeterminada del repositorio de origen se comportarán de forma un poco diferente a las reglas de bifurcación que se aplican a otras ramas. Cualquier regla que se aplique a la rama predeterminada se aplicará automáticamente a cualquier

rama que especifiques como la rama predeterminada. La rama que se estableció anteriormente como la rama predeterminada conservará las reglas que se le apliquen, excepto que ya no estará protegida contra la eliminación. Esa protección solo se aplica a la rama predeterminada actual.

Las reglas de sucursal tienen dos estados: Estándar y Personalizado. El estándar indica que las acciones permitidas en una sucursal son las que coinciden con los permisos del rol que desempeña el usuario en las acciones CodeCatalyst de la sucursal. Para obtener más información sobre qué roles tienen qué permisos, consulte [Trabajar con roles en Amazon CodeCatalyst](#). La opción Personalizado indica que una o más acciones de sucursal tienen acciones que tienen una lista específica de roles permitidos para realizar esa acción y que difieren de los permisos predeterminados que otorga el rol de un usuario en el proyecto.

Note

Si creas una regla de rama para restringir una o más acciones de una rama, la acción Eliminar la rama se establece automáticamente para permitir que solo los usuarios con el rol de administrador del proyecto eliminen esa rama.

En la siguiente tabla se enumeran las acciones y la configuración predeterminada de los roles que pueden realizar estas acciones en una sucursal.

Acciones y funciones de la sucursal

Acción de ramificación	Los roles pueden realizar esta acción cuando no se aplica ninguna regla de ramificación
Fusiona con la rama (esto incluye fusionar una solicitud de extracción con la sucursal)	Administrador del proyecto, colaborador
Dirígete a la sucursal	Administrador del proyecto, colaborador
Elimine la rama	Administrador del proyecto, colaborador
Eliminar la rama (rama predeterminada)	No permitido

No puedes eliminar las reglas de ramificación, pero puedes actualizarlas para permitir acciones de todos los roles que estarían autorizados a realizar esta acción en una rama, lo que elimina la regla de manera efectiva.

Note

Debe tener el rol de administrador de proyectos para configurar las reglas de ramificación de los repositorios de origen de los CodeCatalyst proyectos. Esto no se aplica a los repositorios enlazados. Los repositorios enlazados no admiten las reglas de ramificación de CodeCatalyst

Para ver y editar las reglas de ramificación de un repositorio

1. Navegue hasta el proyecto en el que reside su repositorio.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee ver las reglas de las sucursales.

3. En la página de información general del repositorio, selecciona Sucursales.
4. En la columna Reglas de sucursal, consulta el estado de las reglas de cada rama del repositorio. El estándar indica que las reglas para la acción de ramificación son las predeterminadas para cualquier rama creada en un repositorio de origen y coinciden con los permisos otorgados a esas funciones en un proyecto. La opción Personalizado indica que una o más acciones de rama tienen reglas que restringen una o más acciones permitidas para esa rama a un conjunto diferente de roles.

Para ver los detalles de las reglas de sucursal de una sucursal, elige la palabra Estándar o Personalizada junto a la sucursal que deseas revisar.

5. Para crear o cambiar una regla de sucursal, selecciona Administrar configuración. En la página de configuración del repositorio de origen, en Reglas de sucursal, selecciona Editar.
6. En Branch, elija el nombre de la rama para la que quiere configurar una regla en la lista desplegable. Para cada uno de los tipos de acciones permitidos, elija los roles a los que desea permitir que realicen esa acción en la lista desplegable y, a continuación, elija Guardar.

Comandos de Git para sucursales

Puedes usar Git para crear, gestionar y eliminar ramas en el clon del repositorio de origen que tengas en tu ordenador (tu repositorio local) o en tus entornos de desarrollo y, a continuación,

confirmar y enviar los cambios a tu repositorio de CodeCatalyst origen (el repositorio remoto). Por ejemplo:

Comandos comunes de Git para sucursales

Enumera todas las ramificaciones en el repositorio local y muestra un asterisco (*) junto a la ramificación actual.	<code>git branch</code>
Extrae información sobre todas las ramas existentes en el repositorio remoto al repositorio local.	<code>git fetch</code>
Enumera todas las ramificaciones del repositorio local y todas las ramificaciones de seguimiento remotas del repositorio local.	<code>git branch -a</code>
Enumera solo las ramificaciones de seguimiento remotas del repositorio local.	<code>git branch -r</code>
Crea una rama en el repositorio local con el nombre de rama especificado. Esta rama no aparecerá en el repositorio remoto hasta que confirmes y publiques el cambio.	<code>git branch <i>branch-name</i></code>
Crea una rama en el repositorio local con el nombre de rama especificado y, a continuación, cambia a esa rama.	<code>git checkout -b <i>branch-name</i></code>
Cambia a otra ramificación del repositorio local con el nombre de la ramificación especificada.	<code>git checkout <i>other-branch-name</i></code>
Envía una rama del repositorio local al repositorio remoto utilizando el apodo especificado por el repositorio local para el repositorio remoto y el nombre de la rama especificada. También configura la información de seguimiento ascendente de la rama en el repositorio local.	<code>git push -u <i>remote-name</i> <i>branch-name</i></code>

Combina los cambios de otra ramificación del repositorio local en la ramificación actual del repositorio local.

```
git merge from-other-branch-name
```

Elimina una ramificación del repositorio local a menos que contenga trabajo que no se ha combinado.

```
git branch -d branch-name
```

Elimina una rama del repositorio remoto con el apodo especificado que el repositorio local tiene para el repositorio remoto y el nombre de la rama especificada. Observe el uso de los dos puntos (:). También puede `--delete` especificarlo como parte del comando.

```
git push remote-name :branch-name
```

```
git push remote-name --delete  
branch-name
```

Para obtener más información, consulta la documentación de Git.

Visualización de sucursales y detalles

Puedes ver información sobre las sucursales remotas de Amazon CodeCatalyst, incluidos los detalles de los archivos, las carpetas y las confirmaciones más recientes de una sucursal específica, en la CodeCatalyst consola de Amazon. También puedes usar los comandos de Git y tu sistema operativo local para ver esta información para las sucursales remotas y locales.

Para ver las sucursales (consola)

1. En la CodeCatalyst consola, navegue hasta el proyecto que contiene el repositorio de origen en el que desea ver las ramas. Elige Código, repositorios de origen y, a continuación, elige el repositorio de origen.
2. Elija el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee ver una rama.

3. Aparece la rama predeterminada del repositorio. Puedes ver una lista de los archivos y carpetas de la rama, información sobre la confirmación más reciente y el contenido del archivo

README.md, si existe en la rama. Para ver la información de una rama diferente, selecciónela en la lista desplegable de sucursales del repositorio.

4. Para ver todas las sucursales de un repositorio, selecciona Ver todo. La página de sucursales muestra información sobre el nombre, la confirmación más reciente y las reglas de cada rama.

Para obtener información sobre cómo usar Git y su sistema operativo para ver las ramas y los detalles, consulte [Comandos comunes de Git para sucursales](#), la documentación de Git y la documentación del sistema operativo.

Trabajar con archivos en Amazon CodeCatalyst

En Amazon CodeCatalyst, un archivo es una información independiente y controlada por versiones que está disponible para ti y otros usuarios del repositorio y la sucursal de origen donde está almacenado el archivo. Puedes organizar los archivos del repositorio con una estructura de directorios. CodeCatalyst rastrea automáticamente todos los cambios confirmados en un archivo. Puede almacenar diferentes versiones de un archivo en distintas ramas del repositorio.

Para añadir o editar varios archivos en un repositorio de código fuente, puedes usar un cliente Git, un entorno de desarrollo o un entorno de desarrollo integrado (IDE). Para añadir o editar un solo archivo, puedes usar la CodeCatalyst consola.

Temas

- [Crear o añadir un archivo](#)
- [Visualización de un archivo](#)
- [Edición de un archivo](#)
- [Cambiar el nombre o eliminar un archivo](#)

Crear o añadir un archivo

Para crear y añadir archivos a un repositorio fuente, puedes usar la CodeCatalyst consola de Amazon, un entorno de desarrollo, un entorno de desarrollo integrado (IDE) conectado o un cliente Git. La CodeCatalyst consola incluye un editor de código para crear archivos. Este editor es una forma cómoda de crear o editar un archivo sencillo, como un archivo README.md, en una rama de un repositorio. Cuando trabajes en más de un archivo, considera la posibilidad de [crear un](#) entorno de desarrollo.

Para crear un entorno de desarrollo a partir de un repositorio de código fuente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
3. Elige el repositorio de código fuente en el que quieres trabajar con el código.
4. Seleccione Crear entorno de desarrollo.
5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Realice una de las acciones siguientes:
 - Selecciona Trabajar en una rama existente y, a continuación, selecciona una rama en el menú desplegable de la rama existente.
 - Selecciona Trabajar en una nueva sucursal, ingresa el nombre de una sucursal en el campo Nombre de la sucursal y elige una rama a partir de la cual crear la nueva rama en el menú desplegable Crear rama desde.
7. Si lo desea, añada un nombre para el entorno de desarrollo o edite su configuración.
8. Seleccione Crear.

Para crear un archivo en la CodeCatalyst consola

1. Navegue hasta el proyecto en el que desee crear un archivo. Para obtener más información sobre cómo navegar a un repositorio, consulte [Visualización de un repositorio de fuentes](#).
2. Elija el nombre del repositorio en la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee crear el archivo.
3. (Opcional) Elija la rama en la que desea crear el archivo, si quiere crear el archivo en una rama diferente a la rama predeterminada.
4. Selecciona Crear archivo.
5. Introduzca el nombre del archivo en Nombre de archivo. Añada el contenido del archivo en el editor.

i Tip

Si desea crear el archivo en una subcarpeta o subdirectorio de la raíz de la rama, incluya esa estructura como parte del nombre del archivo.

Cuando esté satisfecho con los cambios, elija Confirmar.

6. En Nombre de archivo, revise el nombre del archivo y realice los cambios que desee. Si lo desea, elija la rama en la que desee crear el archivo de la lista de sucursales disponibles en Branch. En el mensaje de confirmación, si lo desea, introduzca una descripción breve pero informativa del motivo por el que ha realizado este cambio. Se mostrará como la información de confirmación básica de la confirmación que añade el archivo al repositorio de origen.
7. Selecciona Confirmar para confirmar y envía el archivo al repositorio de origen.

También puedes añadir archivos a un repositorio de origen clonándolo en tu ordenador local y utilizando un cliente Git o un entorno de desarrollo integrado (IDE) conectado para enviar los archivos y los cambios.

i Note

Si quieres añadir un submódulo de Git, debes usar un cliente Git o un entorno de desarrollo y ejecutar el `git submodule add` comando. No puedes añadir ni ver los submódulos de Git en la CodeCatalyst consola ni ver las diferencias entre los submódulos de Git en las solicitudes de cambios. Para obtener más información sobre los submódulos de Git, consulta la [documentación de Git](#).

Para añadir un archivo mediante un cliente Git o un entorno de desarrollo integrado (IDE) conectado

1. Clona tu repositorio de código fuente en tu ordenador local. Para obtener más información, consulte [Clonación de un repositorio fuente](#).
2. Cree archivos en su repositorio local o cópielos en su repositorio local.
3. Cree e inserte una confirmación mediante una de las siguientes acciones:
 - Si utilizas un cliente Git, ejecuta el comando en la terminal o en la línea de `git add` comandos especificando los nombres de los archivos que quieres añadir. Como alternativa, para añadir

todos los archivos añadidos o modificados, ejecuta el git add comando seguido de un punto único o doble para indicar si deseas incluir todos los cambios en el nivel de directorio actual (punto único) o todos los cambios en el directorio actual y todos los subdirectorios (punto doble). Para confirmar los cambios, ejecute el git commit -m comando y proporcione un mensaje de confirmación. Para insertar los cambios en el repositorio de origen CodeCatalyst, ejecuta git push. Para obtener más información sobre los comandos de Git, consulte la documentación de Git y [Comandos de Git para sucursales](#).

- Si utilizas un entorno de desarrollo o un IDE, crea archivos y añádelos al IDE y, a continuación, confirma e inserta los cambios. Para obtener más información, consulte [Entornos de desarrollo en CodeCatalyst](#) o consulte la documentación del IDE.

Visualización de un archivo

Puedes ver los archivos de tu repositorio de origen en la CodeCatalyst consola de Amazon. Puedes ver los archivos en la rama predeterminada y en cualquier otra rama. El contenido de los archivos puede variar en función de la rama que elija ver.

Para ver los archivos en la CodeCatalyst consola

1. Navegue hasta el proyecto en el que desee ver los archivos. Para obtener más información, consulte [Visualización de un repositorio de fuentes](#).
2. Elija el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee ver los archivos.
3. Aparece una lista de archivos y carpetas para la rama predeterminada. Los archivos se indican con un icono de papel, mientras que las carpetas se indican con un icono de carpeta.
4. Realice uno de los siguientes procedimientos:
 - Para ver los archivos y carpetas de una rama diferente, selecciónela en la lista de sucursales.
 - Para expandir una carpeta, selecciónela de la lista.
5. Para ver el contenido de un archivo específico, selecciónelo de la lista. El contenido del archivo se mostrará en la rama. Para ver el contenido del archivo en una rama diferente, elija la rama que desee en el selector de ramas.

Tip

Al ver el contenido de un archivo, puede elegir archivos adicionales para verlos en Ver archivos. Para editar un archivo, selecciona Editar.

Puede ver varios archivos en la consola. También puede ver los archivos que ha clonado en su equipo local mediante un cliente Git o un entorno de desarrollo integrado (IDE). Para obtener más información, consulta la documentación de tu cliente Git o IDE.

Note

No puedes ver los submódulos de Git en la CodeCatalyst consola. Para obtener más información sobre los submódulos de Git, consulta la [documentación de Git](#).

Edición de un archivo

Puedes editar archivos individuales en la CodeCatalyst consola de Amazon. Para editar varios archivos a la vez, crea un entorno de desarrollo o clona el repositorio y realiza los cambios mediante un cliente Git o un entorno de desarrollo integrado (IDE). Para obtener más información, consulte [Entornos de desarrollo en CodeCatalyst](#) o [Clonación de un repositorio fuente](#).

Para editar un archivo en la CodeCatalyst consola

1. Navegue hasta el proyecto en el que desee editar un archivo. Para obtener más información sobre cómo navegar a un repositorio, consulte [Visualización de un repositorio de fuentes](#).
2. Elija el repositorio en el que desee editar el archivo. Selecciona Ver sucursales y, a continuación, elige la rama en la que quieres trabajar. Elige el archivo de la lista de archivos y carpetas de esa rama.

Se muestra el contenido del archivo.

3. Elija Editar.
4. En el editor, edite el contenido del archivo y, a continuación, elija Confirmar. Si lo desea, en Confirmar cambios, añada más información sobre el cambio en el mensaje de confirmación. Cuando esté satisfecho con los cambios, elija Confirmar.

Cambiar el nombre o eliminar un archivo

Puedes cambiar el nombre de los archivos o eliminarlos en un entorno de desarrollo, de forma local en tu ordenador o en un entorno de desarrollo integrado (IDE). Una vez que haya cambiado el nombre de los archivos o los haya eliminado, confirme y coloque los cambios en el repositorio de origen. No puedes cambiar el nombre ni eliminar archivos en la CodeCatalyst consola de Amazon.

Trabajar con solicitudes de cambios en Amazon CodeCatalyst

Una solicitud de extracción es la principal forma en que tú y otros miembros del proyecto podéis revisar, comentar y combinar los cambios de código de una rama a otra. Puedes usar las solicitudes de extracción para revisar los cambios de código de forma colaborativa para detectar cambios o correcciones menores, incorporaciones de funciones importantes o nuevas versiones del software que hayas publicado. Si utilizas los problemas para hacer un seguimiento del trabajo en tu proyecto, puedes vincular problemas específicos a las solicitudes de extracción para ayudarte a hacer un seguimiento de los problemas que se están abordando con los cambios de código en la solicitud de cambios de código. Al crear, actualizar, comentar, fusionar o cerrar una solicitud de extracción, se envía automáticamente un correo electrónico al autor de la solicitud de extracción, así como a los revisores necesarios u opcionales de la solicitud de extracción.

Tip

Puedes configurar los eventos de solicitud de extracción sobre los que recibirás correos electrónicos como parte de tu perfil. Para obtener más información, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).

Las solicitudes de extracción requieren dos sucursales en un repositorio de origen: una rama de origen que contiene el código que quieres revisar y una rama de destino, donde quieres fusionar el código revisado. La ramificación de origen contiene la confirmación AFTER, que es la confirmación que contiene los cambios que se desean combinar en la ramificación de destino. La ramificación de destino contiene la confirmación BEFORE, que representa el estado del código antes de que la ramificación de la solicitud de extracción se combine con la ramificación de destino.

Note

Al crear una solicitud de extracción, la diferencia que se muestra es la diferencia entre la punta de la rama de origen y la punta de la rama de destino. Una vez que hayas creado la

solicitud de extracción, la diferencia que se mostrará será entre la revisión de la solicitud de atracción que hayas elegido y la confirmación que figuraba en la punta de la sucursal de destino cuando creaste la solicitud de extracción. Para obtener más información sobre las diferencias y las bases de fusión en Git, consulta [git-merge-base](#) la documentación de Git.

Si bien se crean solicitudes de extracción para un repositorio fuente y ramas específicos, puedes crearlas, verlas, revisarlas y cerrarlas como parte del trabajo con tu proyecto. No tienes que ver el repositorio de origen para poder ver las solicitudes de extracción y trabajar con ellas. El estado de una solicitud de extracción se establece en Abierto al crearla. La solicitud de extracción permanece abierta hasta que la combines en la CodeCatalyst consola, lo que cambia el estado a Combinado, o hasta que lo cierres, lo que cambia el estado a Cerrado.

Cuando hayas revisado tu código, puedes cambiar el estado de la solicitud de extracción de varias maneras:

- Combina la solicitud de extracción en la CodeCatalyst consola. El código de la rama fuente de la solicitud de extracción se fusionará con la rama de destino. El estado de la solicitud de extracción cambiará a Combinado. No se puede volver a cambiar a Abierto.
- Combina las ramas de forma local, inserta los cambios y, a continuación, cierra la solicitud de extracción en la CodeCatalyst consola.
- Usa la CodeCatalyst consola para cerrar la solicitud de extracción sin fusionarla. Esto cambiará el estado a Cerrado y no fusionará el código de la rama de origen con el de destino.

Antes de crear una solicitud de extracción:

- Confirma y envía los cambios de código que deseas revisar a una rama (la rama de origen).
- Configura las notificaciones para tu proyecto, de forma que otros usuarios puedan recibir notificaciones sobre cualquier flujo de trabajo que se ejecute al crear una solicitud de incorporación de cambios. (Este paso es opcional pero se recomienda).

Temas

- [Creación de una solicitud de extracción](#)
- [Visualización de solicitudes de cambios](#)
- [Gestione los requisitos para fusionar una solicitud de extracción con las reglas de aprobación](#)

- [Revisión de una solicitud de extracción](#)
- [Actualización de una solicitud de extracción](#)
- [Fusión de una solicitud de extracción](#)
- [Cerrar una solicitud de extracción](#)

Creación de una solicitud de extracción

Las funciones de IA generativa de Amazon CodeCatalyst se encuentran en versión preliminar y están sujetas a cambios. Solo están disponibles en la región de EE. UU. Oeste (Oregón). El acceso a las funciones generativas de IA varía según el nivel. Para más información, consulte [Precios de](#) .

La creación de solicitudes de extracción ayuda a otros usuarios a ver y revisar los cambios en el código antes de combinarlos en otra ramificación. Primero debe crear una ramificación para los cambios de código. Esta es la ramificación de origen de una solicitud de extracción. Tras confirmar los cambios e insertarlos en el repositorio, puedes crear una solicitud de incorporación de cambios que compare el contenido de la rama de origen con el contenido de la rama de destino.

Puedes crear una solicitud de extracción en la CodeCatalyst consola de Amazon desde una sucursal específica, desde la página de solicitudes de extracción o desde la descripción general del proyecto. Al crear una solicitud de extracción desde una rama específica, se proporciona automáticamente el nombre del repositorio y la rama de origen en la página de creación de la solicitud de extracción. Cuando creas una solicitud de extracción, recibirás automáticamente correos electrónicos sobre cualquier actualización de la solicitud de extracción, así como cuando la solicitud de extracción se fusione o cierre.

Note

Al crear una solicitud de extracción, la diferencia que se muestra es la diferencia entre la punta de la rama de origen y la punta de la sucursal de destino. Una vez creada la solicitud de extracción, la diferencia que se mostrará será entre la revisión de la solicitud de extracción que hayas elegido y la confirmación que estaba en la punta de la sucursal de destino cuando creaste la solicitud de extracción. Para obtener más información sobre las diferencias y las bases de fusión en Git, consulta [git-merge-base](#) la documentación de Git.

Puedes usar la función **Escribe una descripción para mí** al crear solicitudes de extracción para que Amazon Q cree automáticamente una descripción de los cambios contenidos en una solicitud de extracción. Al elegir esta opción, Amazon Q analiza las diferencias entre la rama de origen que contiene los cambios de código y la rama de destino en la que desea combinar estos cambios. A continuación, crea un resumen de cuáles son esos cambios, así como su mejor interpretación de la intención y el efecto de esos cambios.

Note

Desarrollado por Amazon Bedrock: AWS implementa la [detección automática de abusos](#). Como las funciones **Escribir una descripción para mí** y **Crear un resumen de contenido** están integradas en Amazon Bedrock, los usuarios pueden aprovechar al máximo los controles implementados en Amazon Bedrock para garantizar la protección, la seguridad y el uso responsable de la inteligencia artificial (IA).

Para crear una solicitud de extracción


1. Vaya a su proyecto.
2. Realice una de las acciones siguientes:
 - En el panel de navegación, selecciona **Código**, selecciona **Solicitudes de extracción** y, a continuación, selecciona **Crear solicitud de extracción**.
 - En la página de inicio del repositorio, selecciona **Más** y, a continuación, selecciona **Crear solicitud de extracción**.
 - En la página del proyecto, selecciona **Crear solicitud de extracción**.
3. En el repositorio de fuentes, asegúrate de que el repositorio de fuentes especificado es el que contiene el código confirmado. Esta opción solo aparece si no creaste la solicitud de extracción desde la página principal del repositorio.
4. En la rama de destino, elige la rama en la que deseas combinar el código después de revisarlo.
5. En la rama Fuente, elige la rama que contiene el código confirmado.
6. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios a entender qué es lo que hay que revisar y por qué.
7. (Opcional) En la descripción de la solicitud de retirada, proporciona información como un enlace a los problemas o una descripción de tus cambios.

 Tip

Puedes elegir Escribir una descripción para que yo genere CodeCatalyst automáticamente una descripción de los cambios incluidos en la solicitud de cambios. Puedes realizar cambios en la descripción generada automáticamente después de añadirla a la solicitud de extracción.

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

8. (Opcional) En Problemas, selecciona Vincular problemas y, a continuación, elige un problema de la lista o introduce su ID. Para desvincular una incidencia, selecciona el icono de desvinculación.
9. (Opcional) En Revisores obligatorios, selecciona Agregar revisores obligatorios. Elija de la lista de miembros del proyecto para añadirlos. Los revisores necesarios deben aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.

 Note

No puedes añadir un revisor como revisor obligatorio y como revisor opcional. No puedes añadirte a ti mismo como revisor.

10. (Opcional) En Revisores opcionales, selecciona Añadir revisores opcionales. Elija de la lista de miembros del proyecto para añadirlos. Los revisores opcionales no tienen que aprobar los cambios como requisito para poder fusionar la solicitud de extracción en la sucursal de destino.
11. Revisa las diferencias entre las sucursales. La diferencia que se muestra en una solicitud de extracción son los cambios entre la revisión en la rama de origen y la base de fusión, que es la confirmación principal de la rama de destino en el momento en que se creó la solicitud de extracción. Si no se muestra ningún cambio, es posible que las ramas sean idénticas o que hayas elegido la misma rama tanto para el origen como para el destino.
12. Cuando estés seguro de que la solicitud de extracción contiene el código y los cambios que deseas revisar, selecciona Crear.


 Note

Después de crear la solicitud de extracción, puedes añadir comentarios. Los comentarios se pueden añadir a la solicitud de extracción o a líneas individuales de

los archivos, así como a la solicitud de extracción en general. Puedes añadir enlaces a recursos, como archivos, utilizando el signo @ seguido del nombre del archivo.

Para crear una solicitud de extracción desde una sucursal

1. Navega hasta el proyecto en el que deseas crear una solicitud de extracción.
2. En el panel de navegación, selecciona Repositorios de origen y, a continuación, elige el repositorio que contiene la rama en la que deseas revisar los cambios de código.
3. Seleccione la flecha desplegable situada junto al nombre de la rama predeterminada y, a continuación, elija la rama que desee de la lista. Para ver todas las ramas de un repositorio, selecciona Ver todas.
4. Seleccione Más y, a continuación, selecciona Crear solicitud de extracción.
5. El repositorio y la rama de origen se preseleccionan automáticamente. En la rama de destino, elige la rama en la que combinarás el código una vez que se haya revisado. En el título de la solicitud de extracción, introduce un título que ayude a otros usuarios del proyecto a entender qué es lo que se debe revisar y por qué. Si lo deseas, puedes incluir más información en la descripción de la solicitud de extracción, por ejemplo CodeCatalyst, pegando un enlace a temas relacionados o añadiendo una descripción de los cambios que has realizado.

 Note

Los flujos de trabajo configurados para ejecutarse en eventos de creación de solicitudes de extracción se ejecutarán una vez creada la solicitud de extracción, si la rama de destino de la solicitud de extracción coincide con una de las ramas especificadas en el flujo de trabajo.

6. Revisa las diferencias entre las sucursales. Si no se muestran cambios, es posible que las sucursales sean idénticas o que haya elegido la misma sucursal tanto para el origen como para el destino.
7. (Opcional) En Problemas, selecciona Vincular problemas y, a continuación, selecciona un problema de la lista o introduce su ID. Para desvincular una incidencia, selecciona el icono de desvinculación.
8. (Opcional) En Revisores obligatorios, selecciona Agregar revisores obligatorios. Elija de la lista de miembros del proyecto para añadirlos. Los revisores necesarios deben aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.

Note

No puedes añadir un revisor, ya que es obligatorio y opcional. No puedes añadirte a ti mismo como revisor.

- (Opcional) En Revisores opcionales, selecciona Añadir revisores opcionales. Elija de la lista de miembros del proyecto para añadirlos. Los revisores opcionales no tienen que aprobar los cambios antes de poder fusionar la solicitud de extracción en la sucursal de destino.
- Cuando estés seguro de que la solicitud de extracción contiene los cambios que deseas revisar e incluye a los revisores necesarios, selecciona Crear.

Si tienes algún flujo de trabajo configurado para que la sucursal coincida con la sucursal de destino en la solicitud de extracción, verás información sobre esas ejecuciones de flujo de trabajo en la descripción general, en el área de detalles de la solicitud de extracción, una vez creada la solicitud de extracción. Para obtener más información, consulte [Añadir un gatillo para pulsar, tirar o programar](#).

Visualización de solicitudes de cambios

Las funciones de IA generativa de Amazon CodeCatalyst se encuentran en versión preliminar y están sujetas a cambios. Solo están disponibles en la región de EE. UU. Oeste (Oregón). El acceso a las funciones generativas de IA varía según el nivel. Para más información, consulte [Precios de](#).

Puedes ver las solicitudes de cambios de un proyecto en la CodeCatalyst consola de Amazon. La página de resumen del proyecto muestra todas las solicitudes de extracción abiertas de un proyecto. Para ver todas las solicitudes de extracción sin importar el estado, navega a la página de solicitudes de extracción de tu proyecto. Al ver una solicitud de extracción, puedes elegir que se cree para ti un resumen de todos los comentarios que se hayan dejado sobre los cambios en la solicitud de extracción.

Note

Desarrollado por Amazon Bedrock: AWS implementa la [detección automática de abusos](#). Como las funciones Escribir una descripción para mí y Crear un resumen de contenido están

integradas en Amazon Bedrock, los usuarios pueden aprovechar al máximo los controles implementados en Amazon Bedrock para garantizar la protección, la seguridad y el uso responsable de la inteligencia artificial (IA).

Para ver las solicitudes de cambios pendientes

1. Navega hasta el proyecto en el que quieres ver las solicitudes de extracción.
2. En la página del proyecto, se muestran las solicitudes de extracción abiertas, incluida la información sobre quién creó la solicitud de extracción, qué repositorio contiene las ramas de la solicitud de extracción y la fecha en que se creó la solicitud de extracción. Puedes filtrar la vista de solicitudes de extracción abiertas por repositorio de origen.
3. Para ver todas las solicitudes de extracción, selecciona Ver todas. Puedes usar los selectores para elegir entre las opciones. Por ejemplo, para ver todas las solicitudes de cambios, selecciona Cualquier estado y Cualquier autor.


Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Solicitudes de extracción y, a continuación, utiliza los selectores para afinar la vista.

4. En la página de solicitudes de extracción, puedes ordenar las solicitudes de extracción por ID, título, estado y más. Para personalizar qué información y cuánta información se muestra en la página de solicitudes de extracción, selecciona el icono de engranaje.
5. Para ver una solicitud de extracción específica, selecciónela de la lista.
6. Para ver el estado de las ejecuciones de flujos de trabajo asociadas a esta solicitud de extracción, si las hay, selecciona Descripción general y revisa la información en el área de detalles de la solicitud de extracción de la solicitud de extracción en Ejecuciones de flujo de trabajo.

Se ejecutará un flujo de trabajo si el flujo de trabajo está configurado con eventos de creación o revisión de solicitudes de extracción y si los requisitos de la sucursal de destino del flujo de trabajo coinciden con los de la sucursal de destino especificada en la solicitud de extracción. Para obtener más información, consulte [Añadir un gatillo para pulsar, tirar o programar](#).


7. Para ver los problemas vinculados, si los hay, selecciona Descripción general y revisa la información en los detalles de la solicitud de extracción en Problemas. Si quieres ver un problema vinculado, elige su ID en la lista.

8. (Opcional) Para crear un resumen de los comentarios dejados sobre los cambios de código en las revisiones de la solicitud de cambios, selecciona Crear resumen de contenido. El resumen no incluirá ningún comentario dejado en la solicitud de extracción general.

 Note


Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

9. Para ver los cambios de código en la solicitud de cambios, selecciona Cambios. Puedes ver rápidamente cuántos archivos tienen cambios en la solicitud de extracción y qué archivos de la solicitud de extracción tienen comentarios, en Archivos cambiados. El número de comentarios que se muestra junto a una carpeta indica el número de comentarios en los archivos de esa carpeta. Amplíe la carpeta para ver el número de comentarios de cada archivo de la carpeta. También puede ver los comentarios que se hayan dejado en líneas de código específicas.

 Note

No todos los cambios de una solicitud de cambios se pueden mostrar en la consola. Por ejemplo, no puedes ver los submódulos de Git en la consola, por lo que no puedes ver las diferencias de un submódulo en una solicitud de cambios. Es posible que algunas diferencias sean demasiado grandes para mostrarlas. Para obtener más información, consulte [Cuotas para los repositorios de fuentes en CodeCatalyst](#) y [Visualización de un archivo](#).

10. Para ver los informes de calidad de esta solicitud de extracción, selecciona Informes.

 Note

Debes configurar un flujo de trabajo para generar informes a fin de que aparezcan en tus solicitudes de extracción de fondos. Para obtener más información, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#).

Gestione los requisitos para fusionar una solicitud de extracción con las reglas de aprobación

Al crear una solicitud de extracción, puedes optar por añadir revisores obligatorios u opcionales a esa solicitud de extracción individual. Sin embargo, también puedes crear requisitos que deben cumplir todas las solicitudes de extracción al fusionarse con una sucursal de destino específica. Estos requisitos se denominan reglas de aprobación. Las reglas de aprobación se configuran para las sucursales de un repositorio. Cuando creas una solicitud de extracción cuya sucursal de destino tiene configurada una regla de aprobación, debes cumplir los requisitos de esa regla, además de las aprobaciones de los revisores necesarios, antes de poder fusionar la solicitud de extracción con esa sucursal. La creación de reglas de aprobación puede ayudarle a mantener los estándares de calidad para las fusiones de sucursales, como la sucursal predeterminada.

Las reglas de aprobación que se aplican a la rama predeterminada de tu repositorio de origen se comportarán de forma un poco diferente a las reglas de aprobación que se aplican a otras sucursales. Cualquier regla que se aplique a la rama predeterminada se aplicará automáticamente a cualquier rama que especifiques como la rama predeterminada. La rama que se estableció anteriormente como la rama predeterminada conservará las reglas que se le hayan aplicado.

Al crear reglas de aprobación, debe tener en cuenta cómo cumplirán esa regla los usuarios del proyecto, tanto en el presente como en el futuro. Por ejemplo, si tienes seis usuarios en tu proyecto y creas una regla de aprobación que requiere cinco aprobaciones antes de que se pueda fusionar con la sucursal de destino, has creado una regla que exige que todos, excepto la persona que creó la solicitud de extracción, aprueben esa solicitud de extracción antes de que se pueda fusionar.

Note

Debe tener el rol de administrador del proyecto para crear y administrar las reglas de aprobación en CodeCatalyst los proyectos. No puede crear reglas de aprobación para los repositorios enlazados.

No puede eliminar las reglas de aprobación, pero puede actualizarlas para que no requieran ninguna aprobación, lo que, de hecho, elimina la regla.

Para ver y editar las reglas de aprobación de las sucursales de destino para las solicitudes de extracción

1. Navega hasta el proyecto en el que reside tu repositorio.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee ver las reglas de aprobación.

3. En la página de información general del repositorio, seleccione Sucursales.
4. En la columna Reglas de aprobación, seleccione Ver para ver el estado de las reglas de cada rama del repositorio.

En Número mínimo de aprobaciones, el número corresponde al número de aprobaciones necesarias para poder fusionar una solicitud de extracción con esa sucursal.

5. Para crear o cambiar una regla de aprobación, selecciona Administrar configuración. En la página de configuración del repositorio de origen, en Reglas de aprobación, seleccione Editar.

Note

Debe tener el rol de administrador del proyecto para editar las reglas de aprobación.

6. En Branch, elija el nombre de la sucursal para la que desea configurar una regla de aprobación en la lista desplegable. En Número mínimo de aprobaciones, introduzca un número y, a continuación, elija Guardar.

Revisión de una solicitud de extracción

Las funciones de IA generativa de Amazon CodeCatalyst se encuentran en versión preliminar y están sujetas a cambios. Solo están disponibles en la región de EE. UU. Oeste (Oregón). El acceso a las funciones generativas de IA varía según el nivel. Para más información, consulte [Precios de](#) .

Puedes usar la CodeCatalyst consola de Amazon para revisar y comentar de forma colaborativa los cambios incluidos en una solicitud de cambios. Puedes añadir comentarios a líneas de código individuales en la diferencia entre la rama de origen y la de destino, o en la diferencia entre las

revisiones de la solicitud de cambios. Puedes optar por crear un resumen de los comentarios dejados sobre los cambios de código en la solicitud de extracción para ayudarte a entender rápidamente los comentarios dejados por otros usuarios. También puedes optar por crear un entorno de desarrollo para trabajar con el código.

Note

Desarrollado por Amazon Bedrock: AWS implementa la [detección automática de abusos](#). Como las funciones Escribir una descripción para mí y Crear un resumen de contenido están integradas en Amazon Bedrock, los usuarios pueden aprovechar al máximo los controles implementados en Amazon Bedrock para garantizar la protección, la seguridad y el uso responsable de la inteligencia artificial (IA).

Tip

Puedes configurar los eventos de pull request sobre los que recibirás correos electrónicos como parte de tu perfil. Para obtener más información, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).

Las solicitudes de extracción muestran cuál es la diferencia entre la revisión de la solicitud de extracción y la confirmación que estaba en la punta de la sucursal de destino cuando creaste la solicitud de extracción. Esto se denomina base de fusión. Para obtener más información sobre las diferencias y las bases de fusión en Git, consulta [git-merge-base](#) la documentación de Git.

Tip

Cuando trabajes en la consola, sobre todo si llevas un tiempo abierta una solicitud de extracción, te recomendamos actualizar tu navegador para asegurarte de que tienes disponible la última versión de una solicitud de extracción antes de empezar a revisarla.

Para revisar una solicitud de extracción en la CodeCatalyst consola


1. Vaya a su proyecto.
2. Para acceder a las solicitudes de extracción, realiza una de las siguientes acciones:

- Si la solicitud de extracción aparece en la página del proyecto, selecciónela de la lista.
 - Si la solicitud de extracción no aparece en la página del proyecto, selecciona Ver todo. Usa los filtros y ordena para encontrar la solicitud de extracción y, a continuación, selecciónela de la lista.
 - En el panel de navegación, selecciona Código y, a continuación, selecciona Solicitudes de extracción.
3. Selecciona de la lista la solicitud de extracción que desees revisar. Puedes filtrar la lista de solicitudes de extracción escribiendo parte de su nombre en la barra de filtros.
 4. En Descripción general, puedes revisar el nombre y el título de la solicitud de extracción. Puedes crear y ver los comentarios que se dejan en la propia solicitud de extracción. También puedes ver los detalles de la solicitud de extracción, incluida la información sobre la ejecución del flujo de trabajo, los problemas relacionados, los revisores, el autor de la solicitud de extracción y las posibles estrategias de fusión.

 Note

Los comentarios que se dejan en líneas de código específicas aparecen en Cambios.

5. (Opcional) Para añadir un comentario que se aplique a toda la solicitud de extracción, expande Comentarios en la solicitud de extracción y, a continuación, selecciona Crear comentario.
6. (Opcional) Para ver un resumen de todos los comentarios dejados sobre los cambios en las revisiones de esta solicitud de extracción, selecciona Crear resumen de comentarios.

 Note


Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas en el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

7. En Cambios, puedes ver las diferencias entre la sucursal de destino y la revisión más reciente de la pull request. Si hay más de una revisión, puedes cambiar las revisiones que se comparan en la diferencia entre ellas. Para obtener más información sobre las revisiones, consulte [Revisiones](#).

 Tip

Puedes ver rápidamente cuántos archivos tienen cambios en la solicitud de extracción y qué archivos de la solicitud de extracción tienen comentarios, en Archivos cambiados. El número de comentarios que se muestra junto a una carpeta indica el número de comentarios en los archivos de esa carpeta. Amplíe la carpeta para ver el número de comentarios de cada archivo de la carpeta.

8. Para cambiar la forma en que se muestran las diferencias, elija entre Unificado y Dividido.
9. Para añadir un comentario a una línea de la solicitud de extracción, dirígete a la línea sobre la que quieres añadir un comentario. Elige el icono de comentario que aparece en esa línea, introduce un comentario y, a continuación, selecciona Guardar.
10. Para ver los cambios entre las revisiones de una solicitud de cambios o entre sus ramas de origen y destino, selecciona una de las opciones disponibles en la sección Comparar. Los comentarios sobre las líneas de las revisiones se conservan en esas revisiones.
11. Si has configurado tu flujo de trabajo para generar un informe de cobertura de códigos sobre los activadores de las solicitudes de extracción, puedes ver los resultados de la cobertura de líneas y sucursales en la solicitud de extracción correspondiente. Para ocultar los resultados de la cobertura del código, selecciona Ocultar la cobertura del código. Para obtener más información, consulte [Informes de cobertura de código](#).
12. Si quieres realizar cambios de código en la solicitud de extracción, puedes crear un entorno de desarrollo a partir de la solicitud de extracción. Seleccione Crear entorno de desarrollo. Si lo desea, agregue un nombre para el entorno de desarrollo o edite su configuración y, a continuación, elija Crear.
13. En Informes, puedes ver los informes de calidad de esta solicitud de extracción. Si hay más de una revisión, puedes cambiar las revisiones que se comparan en la diferencia entre ellas. Puede filtrar los informes por nombre, estado, flujo de trabajo, acción y tipo.

 Note

Debes configurar un flujo de trabajo para generar informes para que aparezcan en tus solicitudes de extracción de cambios. Para obtener más información, consulte [Configuración de informes](#).

14. Para ver un informe específico, selecciónalo de la lista. Para obtener más información, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#).
15. Si apareces como revisor de esta solicitud de información y quieres aprobar los cambios, asegúrate de que estás viendo la revisión más reciente y, a continuación, selecciona Aprobar.

 Note

Todos los revisores necesarios deben aprobar una solicitud de extracción antes de que se pueda fusionar.

Actualización de una solicitud de extracción

Puedes facilitar que otros miembros del proyecto revisen el código actualizando la solicitud de extracción. Puedes actualizar una solicitud de extracción para cambiar sus revisores, sus enlaces a los números, el título de la solicitud de extracción o su descripción. Por ejemplo, es posible que desees cambiar los revisores necesarios para que una solicitud de selección elimine a una persona que esté de vacaciones y añadir a otra persona. También puedes actualizar una solicitud de cambios de código con más cambios de código enviando las confirmaciones a la rama fuente de una solicitud de cambios abierta. Cada vez que se envía a la rama fuente de una solicitud de incorporación de cambios en el repositorio de CodeCatalyst fuentes, se crea una revisión. Los miembros del proyecto pueden ver las diferencias entre las revisiones en una solicitud de extracción.

Para actualizar los revisores de una solicitud de cambios

1. Navega hasta el proyecto en el que deseas actualizar los revisores de una solicitud de extracción.
2. En la página del proyecto, en Abrir solicitudes de extracción, selecciona la solicitud de extracción en la que quieres actualizar a los revisores. Como alternativa, en el panel de navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, elige la solicitud de extracción que deseas actualizar.
3. (Opcional) En Descripción general, en el área de detalles de la solicitud de retirada, selecciona el signo más para añadir revisores obligatorios u opcionales. Selecciona la X situada junto a un revisor para eliminarlo como revisor opcional u obligatorio.
4. (Opcional) En Descripción general, en el área de detalles de la solicitud de extracción, selecciona Vincular problemas para vincular un problema a la solicitud de extracción y, a

continuación, selecciona un tema de la lista o introduce su ID. Para desvincular una incidencia, selecciona el icono de desvinculación situado junto a la incidencia que quieres desvincular.

Para actualizar los archivos y el código en la rama fuente de una solicitud de incorporación de cambios

1. Para actualizar varios archivos, [crea un entorno de desarrollo](#) o clona el repositorio y su rama fuente y usa un cliente Git o un entorno de desarrollo integrado (IDE) para realizar cambios en los archivos de la rama fuente. Confirma y envía los cambios a la rama fuente del repositorio CodeCatalyst fuente para actualizar automáticamente la solicitud de cambios con los cambios. Para obtener más información, consulte [Clonación de un repositorio fuente](#) y [Trabajar con compromisos en Amazon CodeCatalyst](#).
2. Para actualizar un archivo individual en una rama de origen, puedes usar un cliente Git o un IDE como lo harías con varios archivos. También puedes editarlo directamente en la CodeCatalyst consola. Para obtener más información, consulte [Edición de un archivo](#).

Para actualizar el título y la descripción de una solicitud de extracción

1. Navega hasta el proyecto en el que deseas actualizar el título o la descripción de una solicitud de extracción.
2. La página del proyecto muestra las solicitudes de extracción abiertas, incluida información sobre quién creó la solicitud de extracción, qué repositorio contiene las ramas de la solicitud de extracción y cuándo se creó la solicitud de extracción. Puedes filtrar la vista de solicitudes de extracción abiertas por repositorio de origen. Elige la solicitud de extracción que quieres cambiar de la lista.
3. Para ver todas las solicitudes de extracción, selecciona Ver todas. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Solicitudes de extracción. Usa el cuadro de filtro o las funciones de ordenación para buscar la solicitud de extracción que deseas cambiar y, a continuación, selecciónela.
4. En Descripción general, selecciona Editar.
5. Cambia el título o la descripción y, a continuación, selecciona Guardar.

Fusión de una solicitud de extracción

Una vez que se haya revisado tu código y todos los revisores necesarios lo hayan aprobado, puedes fusionar una solicitud de extracción en la CodeCatalyst consola utilizando una estrategia de fusión compatible, como la aceleración. No todas las estrategias de fusión compatibles con la CodeCatalyst consola están disponibles como opciones para todas las solicitudes de extracción. CodeCatalyst evalúa la fusión y solo permite elegir entre las estrategias de fusión que están disponibles en la consola y que permiten fusionar la rama de origen con la de destino. También puedes combinar una solicitud de cambios con las estrategias de fusión de Git que elijas ejecutando el `git merge` comando en tu ordenador local o en un entorno de desarrollo para fusionar la rama de origen con la de destino. A continuación, puedes enviar esos cambios de la rama de destino al repositorio de origen CodeCatalyst.

Note

Si se fusiona la rama y se insertan los cambios en Git, no se cierra automáticamente la solicitud de cambios.

Si tienes el rol de administrador del proyecto, también puedes optar por fusionar una solicitud de extracción que aún no cumpla con todos los requisitos de aprobación y las normas de aprobación.

Fusionar una solicitud de extracción (consola)


Puedes fusionar una solicitud de extracción en la CodeCatalyst consola si no hay conflictos de fusión entre las sucursales de origen y destino y si todos los revisores necesarios han aprobado la solicitud de extracción. Si hay conflictos o si la fusión no se puede completar, el botón de fusión está inactivo y aparece la etiqueta No se puede combinar. En ese caso, debe obtener la aprobación de los aprobadores necesarios, resolver los conflictos localmente si es necesario e introducir esos cambios antes de poder realizar la fusión. Al fusionar una solicitud de extracción, se enviará automáticamente un correo electrónico al creador de la solicitud de extracción, así como a los revisores necesarios u opcionales. No se cerrará ni cambiará automáticamente el estado de ningún problema relacionado con la solicitud de extracción.

 Tip

Puedes configurar los eventos de solicitud de cambios sobre los que recibirás correos electrónicos como parte de tu perfil. Para obtener más información, consulte [Administrar las notificaciones en Amazon CodeCatalyst](#).

Para fusionar una solicitud de cambios

1. Navega hasta el proyecto en el que deseas fusionar una solicitud de extracción.
2. En la página del proyecto, en Abrir solicitudes de extracción, elige la solicitud de extracción que deseas fusionar. Si no ves la solicitud de extracción, selecciona Ver todas las solicitudes de extracción y, a continuación, selecciónela de la lista. Como alternativa, en el panel de navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, elige la solicitud de extracción que deseas fusionar. Elija Merge (Combinar).
3. Elige entre las estrategias de fusión disponibles para la solicitud de extracción. Si lo deseas, selecciona o deselecciona la opción para eliminar la rama de origen después de fusionar la solicitud de extracción y, a continuación, selecciona Fusionar.

 Note

Si el botón Fusionar está inactivo o ves la etiqueta No se puede combinar, significa que los revisores necesarios aún no han aprobado la solicitud de extracción o la solicitud de extracción no se puede fusionar en la consola. CodeCatalyst Si un revisor no ha aprobado una solicitud de extracción, aparece el icono de un reloj en el área de detalles de la solicitud de extracción de información en la sección Descripción general. Si todos los revisores necesarios han aprobado la solicitud de extracción, pero el botón de combinación sigue inactivo, es posible que se esté produciendo un conflicto de fusión. Selecciona la etiqueta No se puede combinar subrayada para ver más detalles sobre por qué la solicitud de extracción no se puede fusionar. Puedes resolver los conflictos de fusión para la rama de destino en un entorno de desarrollo o en la CodeCatalyst consola y, a continuación, fusionar la solicitud de extracción, o puedes resolver los conflictos y fusionar localmente y, después, enviar la confirmación que contiene la fusión a la rama de origen. CodeCatalyst Para obtener más información, consulta [Fusionar una solicitud de cambios \(Git\)](#) la documentación de Git.

Anule los requisitos de fusión

Si tienes el rol de administrador de proyectos, puedes optar por fusionar una solicitud de extracción que aún no cumpla con todos los requisitos para las aprobaciones y normas de aprobación requeridas. Esto se conoce como anular los requisitos de una solicitud de extracción de fondos. Puedes optar por hacerlo si el revisor requerido no está disponible o si surge la necesidad urgente de fusionar una solicitud de extracción específica en una sucursal que tenga reglas de aprobación que no se puedan cumplir rápidamente.

Para fusionar una solicitud de extracción

1. En la solicitud de extracción en la que quieres anular los requisitos y fusionar, selecciona la flecha desplegable situada junto al botón Combinar. Seleccione Anular los requisitos de aprobación.
2. En Motivo de anulación, proporciona detalles sobre por qué estás fusionando esta solicitud de extracción sin cumplir con las normas de aprobación ni con los requisitos de revisión requeridos. Si bien esto es opcional, es muy recomendable.
3. Si lo desea, elija una estrategia de fusión o acepte la predeterminada. También puedes optar por actualizar el mensaje de confirmación generado automáticamente con más detalles.
4. Selecciona o deselecciona la opción para eliminar la rama de origen al fusionar. Te recomendamos que mantengas la rama de origen a la hora de anular los requisitos para fusionar una solicitud de extracción hasta que hayas tenido la oportunidad de revisar la decisión con otros miembros del equipo.
5. Elija Merge (Combinar).

Fusionar una solicitud de cambios (Git)

Git admite muchas opciones para fusionar y administrar sucursales. Los siguientes comandos son algunas de las opciones que puedes usar. Para obtener más información, consulta la documentación disponible en el sitio [web de Git](#). Una vez que hayas fusionado e introducido los cambios, cierra manualmente la solicitud de extracción. Para obtener más información, consulte [Cerrar una solicitud de extracción](#).

Comandos comunes de Git para fusionar ramas

Combina los cambios de la rama de origen del repositorio local con la rama de destino del repositorio local.

```
git checkout destination-branch-name
```

```
git merge source-branch-name
```

Fusiona la rama de origen con la de destino, especificando una fusión rápida. Esto fusiona las ramas y mueve el puntero de la rama de destino a la punta de la rama de origen.

```
git checkout destination-branch-name
```

```
git merge --ff-only source-branch-name
```

Fusiona la rama de origen con la rama de destino, especificando una combinación rápida. Esto combina todas las confirmaciones de la rama de origen en una sola confirmación de fusión en la rama de destino.

```
git checkout destination-branch-name
```

```
git merge --squash source-branch-name
```

Fusiona la rama de origen con la de destino, especificando una fusión a tres bandas. Esto crea una confirmación de fusión y agrega las confirmaciones individuales de la rama de origen a la rama de destino.

```
git checkout destination-branch-name
```

```
git merge --no-ff source-branch-name
```

Elimina la rama de origen del repositorio local. Esto es útil para limpiar tu repositorio local después de fusionarlo con la rama de destino y enviar los cambios al repositorio de origen.

```
git branch -d source-branch-name
```

Elimina la rama de origen del repositorio remoto (el repositorio de origen CodeCatalyst) utilizando el apodo especificado por el repositorio local para el repositorio remoto. Observe el uso de los dos puntos (:). También puede `--delete` especificarlo como parte del comando.

```
git push remote-name :source-branch-name
```

```
git push remote-name --delete source-branch-name
```

Cerrar una solicitud de extracción

Puedes marcar una solicitud de extracción como cerrada. Esto no fusiona la solicitud de extracción, pero puede ayudarte a determinar qué solicitudes de extracción requieren una acción y qué solicitudes de extracción ya no son relevantes. Te recomendamos cerrar una solicitud de extracción después de fusionarla. Al cerrar una solicitud de extracción, se enviará automáticamente un correo electrónico al creador de la solicitud de extracción, así como a cualquier revisor obligatorio u opcional. No cambiará automáticamente el estado de ningún problema relacionado con la solicitud de extracción.

Note

No puedes volver a abrir una solicitud de cambios después de haberla cerrado.

Para cerrar una solicitud de extracción

1. Navega hasta el proyecto en el que quieres cerrar una solicitud de extracción.
2. En la página del proyecto, se muestran las solicitudes de extracción abiertas. Elige la solicitud de extracción que quieres cerrar.
3. Elija Close.
4. Revisa la información y, a continuación, selecciona Cerrar solicitud de extracción.

Trabajar con compromisos en Amazon CodeCatalyst

Las confirmaciones son instantáneas del contenido y de los cambios en el contenido de su repositorio. Cada vez que un usuario confirma y envía un cambio a una sucursal, esa información se guarda. La información de confirmación de Git incluye el autor de la confirmación, la persona que realizó el cambio, la fecha y la hora y los cambios realizados. Cuando creas o editas un archivo en la CodeCatalyst consola de Amazon, se incluye automáticamente información similar, pero el nombre del autor es tu nombre CodeCatalyst de usuario. También puedes añadir etiquetas Git a las confirmaciones para ayudarte a identificar confirmaciones específicas.

En Amazon CodeCatalyst, puedes:

- Ver una lista de confirmaciones de una sucursal.

- Consulta las confirmaciones individuales, incluidos los cambios realizados en una confirmación en comparación con la confirmación principal o las principales.

También puedes ver archivos y carpetas. Para obtener más información, consulte [Trabajar con archivos en Amazon CodeCatalyst](#).

Temas

- [Visualización de las confirmaciones a una sucursal](#)
- [Cambiar la forma en que se muestran las confirmaciones \(CodeCatalystconsola\)](#)

Visualización de las confirmaciones a una sucursal

Para ver el historial de los cambios realizados en una sucursal, consulta las confirmaciones de la sucursal en la CodeCatalyst consola. Esto te ayuda a entender quién realizó cambios en la sucursal y cuándo. También puedes revisar los cambios realizados en una confirmación específica.

También puedes ver las confirmaciones mediante tu cliente Git. Para obtener más información, consulta la documentación de Git.

Para ver las confirmaciones (consola)

1. Navega hasta el proyecto que contiene el repositorio de origen en el que quieres ver las confirmaciones.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elige el repositorio en el que quieres ver las confirmaciones en una rama.

3. Se muestra la rama predeterminada del repositorio, que incluye información sobre la confirmación más reciente en la rama. Selecciona Confirmaciones. Como alternativa, selecciona Más y, a continuación, selecciona Ver confirmaciones.
4. Para ver las confirmaciones de otra rama, selecciona el selector de ramas y, a continuación, elige el nombre de la rama.
5. Para ver los detalles de una confirmación en particular, elige su título en Título de la confirmación. Se muestran los detalles de la confirmación, incluida la información sobre su

confirmación principal y los cambios realizados en el código, comparando la confirmación principal con la confirmación especificada.

Tip

Si una confirmación tiene más de una confirmación principal, puedes elegir de qué confirmación principal quieres ver la información y mostrar los cambios pulsando el icono desplegable situado junto al ID de confirmación principal.

Cambiar la forma en que se muestran las confirmaciones (CodeCatalystconsola)

Puedes cambiar la información que se muestra en la vista de confirmaciones. Puedes optar por ocultar o mostrar columnas como el autor y el ID de confirmación.

Para cambiar la forma en que se muestran las confirmaciones (consola)

1. Navega hasta el proyecto que contiene el repositorio de origen en el que quieres ver las confirmaciones.
2. Elige el nombre del repositorio de la lista de repositorios de origen del proyecto. Como alternativa, en el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.

Elija el repositorio en el que desee cambiar la forma en que ve las confirmaciones.

3. Se muestra la rama predeterminada del repositorio, que incluye información sobre la confirmación más reciente en la rama. Selecciona Confirmaciones.
4. Seleccione el icono de la rueda.
5. En Preferencias, elige el número de confirmaciones que deseas mostrar y elige si deseas mostrar información sobre el autor de las confirmaciones, la fecha de las confirmaciones y el ID de las confirmaciones.

Note

No puedes ocultar el título de la confirmación en la pantalla de información.



6. Cuando haya realizado los cambios, seleccione Guardar para guardarlos o Cancelar para descartarlos.

Cuotas para los repositorios de fuentes en CodeCatalyst

En la siguiente tabla, se describen las cuotas y los límites de los repositorios de origen en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Recurso	Información
Nombres de ramificaciones	<p>Cualquier combinación de caracteres permitida de entre 1 y 256 caracteres de longitud y debe ser única en un repositorio. Los nombres de ramificaciones no pueden:</p> <ul style="list-style-type: none"> • Empezar o terminar con una barra inclinada (/) o un punto (. • Estar formados únicamente por el carácter @. • Contener dos o más puntos (..) o barras inclinadas (//) consecutivos o la siguiente combinación de caracteres: @{ • contener espacios o alguno de los siguientes caracteres: ? ^ * [\ ~ : <p>Los nombres de ramificaciones son referencias. Muchas de las limitaciones de los nombres de ramificaciones se basan en el estándar de referencia de Git. Para obtener más información, consulte Git Internals y git-check-ref-format.</p>
Comentarios sobre una solicitud de cambios	Máximo de 1000 en una solicitud de extracción.
Confirma un mensaje	Máximo de 1024 caracteres.

Recurso	Información
Rutas de archivo	<p>Cualquier combinación de caracteres permitidos de entre 1 y 4,096 caracteres. Las rutas de archivo deben ser un nombre inequívoco que especifica el archivo y la ubicación exacta del archivo. Las rutas de archivo no pueden tener una profundidad superior a los 20 directorios. Además, las rutas de archivo no pueden:</p> <ul style="list-style-type: none"> • contener cadenas vacías • ser una ruta de archivo relativa • incluir ninguna de las siguientes combinaciones de caracteres: <ul style="list-style-type: none"> <code>./</code> <code>../</code> <code>//</code> • terminar con una barra diagonal final o barra inversa <p>Las rutas y los nombres de archivo deben estar totalmente especificados. El nombre y la ruta a un archivo de su equipo local deben cumplir los estándares de ese sistema operativo. Al especificar la ruta a un archivo en un repositorio, utilice los estándares de Amazon Linux.</p>
Tamaño del archivo	Un máximo de 6 MB para cada archivo individual cuando utilice la CodeCatalyst consola.
Tamaño del archivo visible en la consola CodeCatalyst	Un máximo de 6 MB para cada archivo individual cuando se utiliza la CodeCatalyst consola.

Recurso	Información
Tamaño de blob de Git	<p>Máximo de 2 GB.</p> <div data-bbox="829 302 1507 905"><p> Note</p><p>No hay límite al número ni al tamaño total de archivos en una única confirmación, siempre que los metadatos no superen los 6 MB y que ningún blob supere los 2 GB por sí solo. Sin embargo, como práctica recomendada, considera la posibilidad de realizar varias confirmaciones más pequeñas en lugar de una confirmación grande.</p></div>
Metadatos para confirmaciones	<p>Un máximo de 6 MB para los metadatos combinados de una confirmación (por ejemplo, la combinación de la información del autor, la fecha, la lista de confirmaciones principal y los mensajes de confirmación).</p> <div data-bbox="829 1209 1507 1675"><p> Note</p><p>No hay límite al número ni al tamaño total de archivos en una única confirmación, siempre que los metadatos no superen los 20 MB y que ningún archivo individual supere los 6 MB y ningún blob supere los 6 GB por sí solo.</p></div>
Número de CodeCatalyst problemas que se pueden vincular a una solicitud de cambios	50

Recurso	Información
Número de incidencias de Jira que se pueden vincular a una solicitud de extracción	50
Número de solicitudes de extracción abiertas en un espacio	Máximo de 1000 para un CodeCatalyst espacio de Amazon.
Número total de solicitudes de cambios en un espacio	Máximo de 10 000 para un CodeCatalyst espacio de Amazon.
Número de referencias en una única inserción	Máximo de 4 000, incluidas crear, eliminar y actualizar. No hay límite para el número total de referencias en el repositorio.
Número de repositorios en un espacio	Un máximo de 5000 por CodeCatalyst espacio en Amazon.
Descripciones de repositorios	Cualquier combinación de caracteres de entre 0 y 1 000 caracteres. Las descripciones de repositorios son opcionales.
Nombres de repositorio	<p>Los nombres de repositorio deben ser únicos dentro de un proyecto. Pueden contener cualquier combinación de letras, números, puntos, guiones bajos y guiones de entre 1 y 100 caracteres de longitud. Los nombres no distinguen entre mayúsculas y minúsculas. Los nombres de los repositorios no pueden terminar en .git, no pueden contener espacios ni ninguno de los siguientes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :</p>

Recurso	Información
Tamaño del repositorio	El tamaño de los repositorios se ve afectado por los límites generales de almacenamiento del espacio. Para obtener más información, consulte Precios de y Solución de problemas con los repositorios de código fuente .
Revisores de una solicitud de extracción	Un máximo de 100 revisores en total (opcional o obligatorio) para una solicitud de selección.
Resúmenes escritos de las solicitudes de selección	La cantidad máxima de resúmenes escritos para las solicitudes de extracción depende del nivel de facturación de tu espacio. Para obtener más información, consulte Precios .

Entornos de desarrollo en CodeCatalyst

Los entornos de desarrollo son entornos de desarrollo basados en la nube. En Amazon CodeCatalyst, utilizas Dev Environments para trabajar en el código almacenado en los repositorios de código fuente de tu proyecto. Al crear un entorno de desarrollo, tiene varias opciones:

- Cree un entorno de desarrollo específico para el proyecto CodeCatalyst para trabajar en el código con un entorno de desarrollo integrado (IDE) compatible.
- Cree un entorno de desarrollo vacío, clone código en él desde un repositorio fuente y trabaje en ese código con un IDE compatible.
- Cree un entorno de desarrollo en un IDE y clone un repositorio de código fuente en el entorno de desarrollo

Un devfile es un archivo YAML de estándar abierto que estandariza sus entornos de desarrollo. En otras palabras, este archivo codifica las herramientas de desarrollo necesarias para su entorno de desarrollo. Como resultado, puede configurar rápidamente un entorno de desarrollo, cambiar de proyecto y replicar la configuración del entorno de desarrollo entre los miembros del equipo. Los entornos de desarrollo minimizan el tiempo que dedicas a crear y mantener un entorno de desarrollo local, ya que utilizan un archivo de desarrollo que configura todas las herramientas que necesitas para codificar, probar y depurar un proyecto determinado.

Las herramientas del proyecto y las bibliotecas de aplicaciones incluidas en su entorno de desarrollo se definen en el archivo de desarrollo del repositorio de código fuente del proyecto. Si no tienes un archivo de desarrollo en tu repositorio de código fuente, se aplica CodeCatalyst automáticamente un archivo de desarrollo predeterminado. Este devfile predeterminado incluye herramientas para los marcos y lenguajes de programación más utilizados. Si el proyecto se creó mediante un plano, se crea automáticamente un devfile mediante CodeCatalyst. [Para obtener más información sobre el devfile, consulte https://devfile.io.](https://devfile.io)

Una vez que haya creado un entorno de desarrollo, solo usted podrá acceder a él. En tu entorno de desarrollo, puedes ver el código de tu repositorio fuente y trabajar en él en un IDE compatible.

De forma predeterminada, un entorno de desarrollo está configurado para tener un procesador de 2 núcleos, 4 GB de RAM y 16 GB de almacenamiento persistente. Si tienes permisos de administrador de Space, puedes cambiar el nivel de facturación de tu espacio para usar diferentes opciones de configuración del entorno de desarrollo y administrar los límites de procesamiento y almacenamiento. Este es un posible flujo de trabajo para un entorno de desarrollo:

Akua Mansa es una nueva desarrolladora en Example Corp. Akua se unió a su nuevo equipo justo antes del lanzamiento de una nueva versión del producto del equipo. El equipo de Akua necesita que trabaje cuanto antes en una nueva función para la próxima versión del producto. Para evitar un proceso de configuración extenso, Akua acepta una invitación para unirse al proyecto de CodeCatalyst su equipo. A continuación, se asigna el problema relacionado y crea un entorno de desarrollo a partir de una rama existente del repositorio de código fuente de su equipo. El entorno de desarrollo de Akua abre el código del repositorio fuente en el IDE que ha elegido. La instancia IDE abierta está conectada a su entorno de desarrollo, que contiene un archivo de desarrollo que se ha identificado y aplicado automáticamente. El devfile especifica todas las herramientas que necesita para empezar. Akua escribe el código de la nueva función del producto, confirma sus cambios de código, los envía a la rama existente y, después, elimina su entorno de desarrollo cuando termina su trabajo. Akua ha contribuido con código al repositorio de código fuente de su nuevo equipo sin necesidad de un largo proceso de configuración.

Creación de un entorno de desarrollo

Puede crear un entorno de desarrollo de varias maneras:

- Cree un entorno de desarrollo CodeCatalyst con un repositorio de CodeCatalyst código fuente o un repositorio de [código enlazado](#) desde las páginas de información general, entornos de desarrollo o repositorios de código fuente
- Cree un entorno de desarrollo vacío CodeCatalyst que no esté conectado a un repositorio de origen desde la página de entornos de desarrollo
- Cree un entorno de desarrollo en el IDE que prefiera y clone cualquier repositorio fuente en el entorno de desarrollo

Puede crear un entorno de desarrollo por rama de un repositorio. Un proyecto puede tener varios repositorios. Los entornos de desarrollo que cree solo se pueden administrar con su CodeCatalyst cuenta, pero puede abrir el entorno de desarrollo y trabajar en él con cualquiera de los IDE compatibles. Debe tener los entornos de desarrollo AWS Toolkit instalados para poder utilizar los entornos de desarrollo en su IDE. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#). De forma predeterminada, los entornos de desarrollo se crean con un procesador de 2 núcleos, 4 GB de RAM y 16 GB de almacenamiento persistente.

Note

Si ha creado un entorno de desarrollo asociado a un repositorio de código fuente, la columna de recursos siempre muestra la rama que especificó al crear este entorno de desarrollo. Esto se aplica incluso si creas otra rama, cambias a otra rama dentro del entorno de desarrollo o clonas un repositorio adicional. Si has creado un entorno de desarrollo vacío, la columna de recursos estará en blanco.

Entornos de desarrollo integrados compatibles para entornos de desarrollo

Puede utilizar los entornos de desarrollo con los siguientes entornos de desarrollo integrados (IDE) compatibles:

- [AWS Cloud9](#)
- [JetBrains IDE](#)
 - [IntelliJ IDEA Ultimate](#)
 - [GoLand](#)
 - [PyCharm Profesional](#)
- [Visual Studio Code](#)


Crear un entorno de desarrollo en CodeCatalyst

Para empezar a trabajar con Dev Environment en CodeCatalyst, autenticáte e inicia sesión con tu [AWS Builder ID](#) o [SSO](#).

Para crear un entorno de desarrollo a partir de una sucursal


1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya al proyecto donde desea crear un entorno de desarrollo.
3. En el panel de navegación, realice una de las siguientes acciones:
 - Seleccione Descripción general y, a continuación, vaya a la sección Mis entornos de desarrollo.
 - Selecciona Código y, a continuación, selecciona Entornos de desarrollo.

- Elija Código, repositorios de código fuente y elija el repositorio para el que quiere crear un entorno de desarrollo.
4. Seleccione Crear entorno de desarrollo.
 5. Elige un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
 6. Seleccione Clonar un repositorio.
 7. Realice una de las acciones siguientes:
 - a. Elija el repositorio que desea clonar, elija Trabajar en la ramificación existente y, a continuación, elija una ramificación del menú desplegable Ramificación existente.

 Note

Si eliges un repositorio de terceros, debes trabajar en una sucursal existente.

- b. Elija el repositorio que desea clonar, elija Trabajar en una nueva ramificación, introduzca el nombre de la ramificación en el campo Nombre de ramificación, y elija una ramificación a partir de la cual crear la ramificación nueva en el menú desplegable Crear ramificación desde.

 Note

Si creas un entorno de desarrollo desde la página de repositorios de código fuente o desde un repositorio de código fuente específico, no necesitas elegir un repositorio. El entorno de desarrollo se creará a partir del repositorio de origen que haya elegido en la página de repositorios de origen.

8. (Opcional) En Alias: opcional, introduce un alias para el entorno de desarrollo.
9. (Opcional) Pulse el botón de edición de la configuración del entorno de desarrollo para editar la configuración de procesamiento, almacenamiento o tiempo de espera del entorno de desarrollo.
10. (Opcional) En Amazon Virtual Private Cloud (Amazon VPC): opcional, selecciona una conexión de VPC que quieras asociar a tu entorno de desarrollo en el menú desplegable.

Si se establece una VPC predeterminada para su espacio, sus entornos de desarrollo se ejecutarán conectados a esa VPC. Puede anular esto asociando una conexión de VPC diferente. Además, tenga en cuenta que los entornos de desarrollo conectados a VPC no son compatibles.

AWS Toolkit

Note

Al crear un entorno de desarrollo con una conexión de VPC, se crea una nueva interfaz de red dentro de la VPC. CodeCatalyst interactúa con esta interfaz mediante el rol de VPC asociado. Además, asegúrese de que el bloque CIDR de IPv4 no esté configurado para el 172.16.0.0/12 rango de direcciones IP.

11. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo.

Para crear un entorno de desarrollo vacío

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya al proyecto donde desea crear un entorno de desarrollo.
3. En el panel de navegación, realice una de las siguientes acciones:
 - Seleccione Descripción general y, a continuación, vaya a la sección Mis entornos de desarrollo.
 - Seleccione Código y, a continuación, seleccione Entornos de desarrollo.
4. Seleccione Crear entorno de desarrollo.
5. Elija un IDE compatible en el menú desplegable. Para obtener más información, consulte [Entornos de desarrollo integrados compatibles para entornos de desarrollo](#).
6. Seleccione Crear un entorno de desarrollo vacío.
7. (Opcional) En Alias: opcional, introduzca un alias para el entorno de desarrollo.
8. (Opcional) Pulse el botón de edición de la configuración del entorno de desarrollo para editar la configuración de procesamiento, almacenamiento o tiempo de espera del entorno de desarrollo.
9. (Opcional) En Amazon Virtual Private Cloud (Amazon VPC): opcional, seleccione una conexión de VPC que quieras asociar a tu entorno de desarrollo en el menú desplegable.

Si se establece una VPC predeterminada para su espacio, sus entornos de desarrollo se ejecutarán conectados a esa VPC. Puede anular esto asociando una conexión de VPC diferente. Además, tenga en cuenta que los entornos de desarrollo conectados a VPC no son compatibles.

AWS Toolkit

Note

Al crear un entorno de desarrollo con una conexión de VPC, se crea una nueva interfaz de red dentro de la VPC. CodeCatalyst interactúa con esta interfaz mediante el rol de VPC asociado. Además, asegúrese de que el bloque CIDR de IPv4 no esté configurado para el 172.16.0.0/12 rango de direcciones IP.

10. Seleccione Crear. Mientras se crea el entorno de desarrollo, la columna de estado del entorno de desarrollo mostrará Iniciando y la columna de estado mostrará Ejecutando una vez se haya creado el entorno de desarrollo.

Note

Crear y abrir un entorno de desarrollo por primera vez puede llevar de uno a dos minutos.

Note

Una vez que se abra el entorno de desarrollo en el IDE, es posible que tengas que cambiar el directorio al repositorio fuente antes de confirmar e insertar los cambios en el código.

Crear un entorno de desarrollo en un IDE

Puedes usar un entorno de desarrollo integrado (IDE) compatible para trabajar con el código almacenado en los repositorios de código fuente de tu proyecto.

- [Trabajando con Amazon CodeCatalyst en AWS Cloud9](#)
- [Amazon CodeCatalyst en VS Code](#)

Para obtener instrucciones sobre la autenticación, consulte [Autenticación y conexión AWS desde su entorno de desarrollo](#) y [Autenticación para Amazon CodeCatalyst](#) en la Guía del AWS Toolkit for Visual Studio Code usuario.

- [Amazon CodeCatalyst en JetBrains](#)

Para obtener instrucciones sobre la autenticación, consulte [Autenticar y conectar JetBrains Gateway con CodeCatalyst](#) en la Guía del AWS Toolkit for JetBrains usuario.

Detención de un entorno de desarrollo

El `/projects` directorio de un entorno de desarrollo almacena los archivos que se extraen del repositorio de origen y el archivo de desarrollo que se utiliza para configurar el entorno de desarrollo. El `/home` directorio, que está vacío al crear el entorno de desarrollo, almacena los archivos que se crean mientras se utiliza el entorno de desarrollo. Todo el contenido de los `/home` directorios `/projects` y directorios de un entorno de desarrollo se almacena de forma permanente, por lo que puedes dejar de trabajar en un entorno de desarrollo si necesitas cambiar a otro entorno de desarrollo, repositorio o proyecto.

Warning

No se agotará el tiempo de espera de un entorno de desarrollo si alguna instancia, incluidos los navegadores web, las consolas remotas y los IDE, permanece conectada. Por lo tanto, asegúrate de cerrar todas las instancias conectadas para evitar incurrir en costes adicionales.

Un entorno de desarrollo se detendrá automáticamente si permanece inactivo durante el tiempo que se seleccionó en los campos de tiempo de espera durante la creación del entorno de desarrollo. Puede detener el entorno de desarrollo antes de que quede inactivo. Si seleccionó Sin tiempo de espera al crear su entorno de desarrollo, el entorno de desarrollo no se detendrá automáticamente. En su lugar, se ejecutará de forma continua.

Warning

Si detienes un entorno de desarrollo asociado a una conexión de VPC eliminada, no se podrá reanudar.

Para detener un entorno de desarrollo desde la página de entornos de desarrollo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee detener un entorno de desarrollo.
3. En el panel de navegación, selecciona Código.
4. Elija Entornos de desarrollo.
5. Seleccione el botón de radio del entorno de desarrollo que desee detener.

6. En el menú Acciones, selecciona Detener.

Note

El uso informático se factura solo mientras el entorno de desarrollo está en ejecución, pero el uso del almacenamiento se factura durante todo el tiempo que el entorno de desarrollo esté en funcionamiento. Detenga su entorno de desarrollo cuando no esté en uso para detener la facturación informática.

Reanudación de un entorno de desarrollo

El `/projects` directorio de un entorno de desarrollo almacena los archivos que se extraen del repositorio de origen y el archivo de desarrollo que se utiliza para configurar el entorno de desarrollo. El `/home` directorio, que está vacío al crear el entorno de desarrollo, almacena los archivos que se crean mientras se utiliza el entorno de desarrollo. Todo el contenido `/projects` y los `/home` directorios de un entorno de desarrollo se almacenan de forma permanente, por lo que puede dejar de trabajar en un entorno de desarrollo si necesita cambiar a otro entorno de desarrollo, repositorio o proyecto y volver a trabajar en su entorno de desarrollo más adelante.

Un entorno de desarrollo se detendrá automáticamente si permanece inactivo durante el tiempo que se seleccionó en los campos de tiempo de espera durante la creación del entorno de desarrollo. Debe cerrar la pestaña del AWS Cloud9 navegador para que el entorno de desarrollo esté inactivo.

Note

El entorno de desarrollo sigue disponible y en funcionamiento incluso si eliminas la rama con la que lo creaste. Si quieres volver a trabajar en un entorno de desarrollo en el que has eliminado la rama, crea una rama nueva e introduce los cambios en ella.

Para reanudar un entorno de desarrollo desde la página de información general

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee reanudar un entorno de desarrollo y vaya a la sección Mis entornos de desarrollo.
3. Selecciona Reanudar en (IDE).

- Para los JetBrains IDE, elija JetBrains Gateway-EAP cuando se le pida que elija una aplicación para abrir el enlace -gateway. JetBrains Elija Abrir enlace para confirmar cuando se le solicite.
- Para el IDE de VS Code, elija VS Code cuando se le pida que elija una aplicación para abrir el enlace de VS Code. Selecciona Abrir enlace para confirmar.

Para reanudar un entorno de desarrollo desde el repositorio de origen

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee reanudar un entorno de desarrollo.
3. En el panel de navegación, selecciona Código.
4. Elija Repositorios de origen.
5. Elija el repositorio de origen que contiene el entorno de desarrollo que desea reanudar.
6. Elija el nombre de la sucursal para ver un menú desplegable de sus sucursales y, a continuación, elija la sucursal.
7. Elige Resume Dev Environment.
 - Para los JetBrains IDE, elija Abrir enlace para confirmar cuando se le pida si desea permitir que este sitio abra el enlace JetBrains -gateway con JetBrains Gateway? .
 - Para el IDE de VS Code, elija Abrir enlace para confirmar cuando se le pregunte si desea permitir que este sitio abra el enlace de VS Code con Visual Studio Code. .

Para reanudar un entorno de desarrollo desde la página de entornos de desarrollo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee reanudar un entorno de desarrollo.
3. En el panel de navegación, selecciona Código.
4. Elija Entornos de desarrollo.
5. En la columna IDE, elija Reanudar en (IDE) para el entorno de desarrollo.
 - En el caso de los JetBrains IDE, selecciona Abrir enlace para confirmar cuando se te pregunte si deseas permitir que este sitio abra el enlace JetBrains -gateway con JetBrains Gateway? .
 - Para el IDE de VS Code, elija Abrir enlace para confirmar cuando se le pregunte si desea permitir que este sitio abra el enlace de VS Code con Visual Studio Code. .

 Note

La reanudación de un entorno de desarrollo puede tardar unos minutos.

Edición de un entorno de desarrollo

Mientras su IDE está en ejecución, puede editar el entorno de desarrollo. Si editas el tiempo de espera de procesamiento o inactividad, tu entorno de desarrollo se reiniciará después de guardar los cambios.


Para editar un entorno de desarrollo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee editar un entorno de desarrollo.
3. En el panel de navegación, selecciona Código.
4. Elija Entornos de desarrollo.
5. Elija el entorno de desarrollo que desee editar.
6. Elija Editar.
7. Realice los cambios que desee en el tiempo de espera de procesamiento o inactividad.
8. Elija Guardar.

Eliminación de un entorno de desarrollo

Cuando haya terminado de trabajar en el contenido almacenado en su entorno de desarrollo, puede eliminarlo. Crea un nuevo entorno de desarrollo para trabajar en contenido nuevo. Si eliminas tu entorno de desarrollo, el contenido persistente se eliminará permanentemente. Antes de eliminar tu entorno de desarrollo, asegúrate de confirmar y enviar los cambios de código al repositorio fuente original del entorno de desarrollo. Una vez que hayas eliminado tu entorno de desarrollo, se detendrá la facturación de cómputo y almacenamiento del entorno de desarrollo.


Tras eliminar el entorno de desarrollo, es posible que la cuota de almacenamiento tarde unos minutos en actualizarse. Si has alcanzado la cuota de almacenamiento, no podrás crear un nuevo entorno de desarrollo durante este tiempo.

 Important

La eliminación de un entorno de desarrollo no se puede deshacer. Tras eliminar un entorno de desarrollo, ya no podrá recuperarlo.

Para eliminar un entorno de desarrollo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee eliminar un entorno de desarrollo.
3. En el panel de navegación, selecciona Código.
4. Elija Entornos de desarrollo.
5. Elija el entorno de desarrollo que desee eliminar.
6. Elija Eliminar.
7. Introduzca **delete** para confirmar la eliminación del entorno de desarrollo.
8. Elija Eliminar.

 Note

Antes de eliminar una conexión de VPC en su espacio, asegúrese de eliminar el entorno de desarrollo asociado a esa VPC.

Incluso si elimina un entorno de desarrollo, es posible que no elimine la interfaz de red de la VPC. Asegúrese de limpiar sus recursos según sea necesario. Si se produce un error al eliminar un entorno de desarrollo conectado a VPC, debe [desconectar la](#) conexión obsoleta y [eliminarla](#) después de confirmar que no se está utilizando.

Conexión a un entorno de desarrollo mediante SSH

Puede conectarse a su entorno de desarrollo mediante SSH para realizar acciones sin limitaciones, como el reenvío de puertos, la carga y descarga de archivos y el uso de otros IDE.

Note

Si quieres seguir usando SSH durante un tiempo prolongado después de cerrar la pestaña o ventana del IDE, asegúrate de establecer un tiempo de espera alto para tu entorno de desarrollo para que no se detenga debido a la inactividad del IDE.

Requisitos previos

- Necesitas uno de los siguientes sistemas operativos:
 - Windows 10 o posterior y OpenSSH habilitado
 - macOS y Bash versión 3 o superior
 - Linux con yum dpkg o gestores de rpm paquetes y Bash versión 3 o superior
- También necesitas la AWS CLI versión 2.9.4 o superior.

Para conectarse a un entorno de desarrollo mediante SSH

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee conectarse a un entorno de desarrollo mediante SSH.
3. En el panel de navegación, selecciona Código.
4. Elija Entornos de desarrollo.
5. Elige un entorno de desarrollo en ejecución al que quieras conectarte mediante SSH.
6. Selecciona Connect via SSH, elige el sistema operativo que desees y haz lo siguiente:
 - Si aún no lo has hecho, pega y ejecuta el primer comando en el terminal que especifiques. El comando descarga un script y ejecuta las siguientes modificaciones en tu entorno local para que puedas conectarte a tu entorno de desarrollo mediante SSH:
 - Instala el complemento [Session Manager](#) para AWS CLI
 - Modifica tu cuenta local AWS Config y añade un CodeCatalyst perfil para que puedas iniciar sesión con el SSO. Para obtener más información, consulte [Configuración para usar el AWS CLI con CodeCatalyst](#).
 - Modifica tu configuración de SSH local y añade la configuración necesaria para conectarte a tu entorno de desarrollo a través de SSH.

- Añade un script al `~/ .aws/codecatalyst-dev-env` directorio que utiliza el cliente SSH para conectarse a su entorno de desarrollo. Este script llama a la [CodeCatalyst StartDevEnvironmentSession API](#) y utiliza un AWS Systems Manager Session Manager complemento para establecer una AWS Systems Manager sesión con su entorno de desarrollo, que el cliente SSH local utiliza para conectarse de forma segura al entorno de desarrollo remoto.
- Inicia sesión en Amazon CodeCatalyst mediante AWS el inicio de sesión único con el segundo comando. [Este comando solicita y recupera las credenciales para que el script del ~/ .aws/codecatalyst-dev-env directorio pueda llamar a la API. CodeCatalyst StartDevEnvironmentSession](#) Este comando debe ejecutarse cada vez que caduquen las credenciales. Cuando ejecutes el último comando en el modal (`ssh<destination>`), aparecerá un error si tus credenciales han caducado o si no has realizado el inicio de sesión único como se indica en este paso.
- Conéctese a su entorno de desarrollo especificado a través de SSH con el tercer comando. Este comando tiene la siguiente estructura:

```
ssh codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

También puedes usar este comando para realizar otras acciones permitidas por el cliente SSH, como el reenvío de puertos o la carga y descarga de archivos:

- Reenvío de puertos:

```
ssh -L <local-port>:127.0.0.1:<remote-port> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

- Cargar un archivo al directorio principal de su entorno de desarrollo:

```
scp -0 </path-to-local-file> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>:</path-to-remote-file-or-directory>
```

Configuración de su entorno de desarrollo

Un devfile es un estándar abierto que le ayuda a personalizar los entornos de desarrollo de su equipo. Un devfile es un archivo YAML que codifica las herramientas de desarrollo necesarias. Al configurar un devfile, puede predeterminar las herramientas de proyecto y las bibliotecas de aplicaciones que necesita y Amazon las CodeCatalyst instala en su entorno de desarrollo por usted.

El archivo de desarrollo es específico del repositorio para el que se creó y puede crear un archivo de desarrollo independiente para cada repositorio. Su entorno de desarrollo admite comandos y eventos, y proporciona una imagen de archivo de desarrollo universal predeterminada.

Si crea un proyecto con el plano vacío, puede crear un devfile manualmente. Si crea un proyecto con un esquema diferente, CodeCatalyst crea un devfile automáticamente. El `/projects` directorio de un entorno de desarrollo almacena los archivos que se extraen del repositorio de origen y del devfile. El `/home` directorio, que está vacío al crear un entorno de desarrollo por primera vez, almacena los archivos que se crean mientras se utiliza el entorno de desarrollo. Todo el contenido de los `/home` directorios `/projects` y de un entorno de desarrollo se almacena de forma persistente.

Note

La `/home` carpeta solo cambia si cambias el nombre del devfile o el nombre del componente devfile. Si cambia el nombre del componente devfile o devfile, se sustituirá el contenido del `/home` directorio y no se podrán recuperar los datos del `/home` directorio anterior.

Si crea un entorno de desarrollo con un repositorio de código fuente que no contiene un archivo de desarrollo en su raíz, o si crea un entorno de desarrollo sin un repositorio de código fuente, se aplicará automáticamente un archivo de desarrollo universal predeterminado al repositorio de origen. Se utiliza la misma imagen de archivo de desarrollo universal predeterminada para todos los IDE. CodeCatalyst actualmente es compatible con la versión 2.0.0 de devfile. Para obtener más información sobre el archivo de desarrollo, consulte el [esquema de archivo de desarrollo: versión 2.0.0](#).

Note

Solo puede incluir imágenes de contenedor público en su archivo de desarrollo.

Tenga en cuenta que los entornos de desarrollo conectados a VPC solo admiten las siguientes imágenes de devfile:

- Imagen universal
- Imágenes privadas de Amazon ECR, si el repositorio se encuentra en la misma región que la VPC

Temas

- [Edición de un archivo de desarrollo de repositorio para un entorno de desarrollo en CodeCatalyst](#)
- [Edición de un archivo de desarrollo del repositorio para un entorno de desarrollo en un IDE](#)
- [Mover un archivo de desarrollo del repositorio a un entorno de desarrollo](#)
- [Modo de recuperación](#)
- [Funciones de Devfile compatibles con CodeCatalyst](#)
- [Ejemplo: configurar el archivo de desarrollo para su entorno de desarrollo](#)
- [Comandos de Devfile](#)
- [Eventos de Devfile](#)
- [Componentes de Devfile](#)
- [Imágenes universales devfile](#)

Edición de un archivo de desarrollo de repositorio para un entorno de desarrollo en CodeCatalyst

Para editar el archivo de desarrollo del repositorio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto que contiene el repositorio de código fuente para el que desea editar el devfile.
3. En el panel de navegación, selecciona Código.
4. Elija Repositorios de origen.
5. Elija el repositorio de origen que contiene el devfile que desea editar.
6. En la lista de archivos, selecciona el `devfile.yaml` archivo.
7. Elija Editar.
8. Edite el archivo de desarrollo.
9. Selecciona Confirmar o crea una solicitud de cambios para que un miembro del equipo pueda revisar y aprobar los cambios.

Note

Si editas tu devfile, tendrás que reiniciarlo para que los cambios surtan efecto. Esto se puede hacer ejecutando `/aws/mde/mde start --location devfile.yaml` Si hay algún

problema al iniciar tu devfile, entrará en modo de recuperación. Sin embargo, si editas un devfile asociado a un entorno de desarrollo conectado a una VPC, tendrás que reiniciar el entorno de desarrollo en su lugar para que los cambios surtan efecto.

Puede revisar qué archivo de desarrollo se está utilizando ejecutando `/aws/mde/mde status` El campo de ubicación tiene la ruta del devfile relativa a la carpeta del `/projects` entorno.

```
{
  "status": "STABLE",
  "location": "devfile.yaml"
}
```

Edición de un archivo de desarrollo del repositorio para un entorno de desarrollo en un IDE

Para cambiar la configuración de un entorno de desarrollo, debe editar el archivo de desarrollo. Le recomendamos que edite el archivo de desarrollo en un IDE compatible y, a continuación, actualice su entorno de desarrollo, pero también puede editar el archivo de desarrollo desde la raíz del repositorio de origen en CodeCatalyst. Si editas el devfile en un IDE compatible, debes confirmar los cambios e insertarlos en el repositorio de origen o crear una solicitud de cambios para que un miembro del equipo pueda revisar y aprobar las modificaciones del devfile.

- [Edición del archivo de desarrollo del repositorio para un entorno de desarrollo en AWS Cloud9](#)
- [Edición del archivo de desarrollo del repositorio para un entorno de desarrollo en VS Code](#)
- [Edición del archivo de desarrollo del repositorio para un entorno de desarrollo en JetBrains](#)

Mover un archivo de desarrollo del repositorio a un entorno de desarrollo

Puedes mover el archivo de desarrollo predeterminado `/projects/devfile.yaml` a tu repositorio de código fuente. Para actualizar la ubicación del devfile, usa el siguiente comando: `/aws/mde/mde start --location repository-name/devfile.yaml`

Modo de recuperación

Si hay algún problema al iniciar el archivo de desarrollo, entrará en modo de recuperación para que pueda seguir conectándose a su entorno y corregir su archivo de desarrollo. Mientras esté en modo

de recuperación, al ejecutarlo `/aws/mde/mde status` no se incluirá la ubicación del archivo de desarrollo.

```
{
  "status": "STABLE"
}
```

Puedes comprobar el error en los registros de abajo `/aws/mde/logs`, corregir el devfile e intentar volver a `/aws/mde/mde start` ejecutarlo.

Funciones de Devfile compatibles con CodeCatalyst

CodeCatalyst admite las siguientes funciones de devfile en la versión 2.0.0. Para obtener más información sobre el archivo de desarrollo, consulte el [esquema de archivo de desarrollo: versión 2.0.0](#).

Característica	Tipo
<code>exec</code>	Comando
<code>postStart</code>	Evento
<code>container</code>	Componente
<code>args</code>	Propiedades de los componentes
<code>env</code>	Propiedades de los componentes
<code>mountSources</code>	Propiedades de los componentes
<code>volumeMounts</code>	Propiedades de los componentes

Ejemplo: configurar el archivo de desarrollo para su entorno de desarrollo

El siguiente es un ejemplo de un archivo de desarrollo simple.

```
schemaVersion: 2.0.0
metadata:
  name: a12
```



```
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      mountSources: true
      command: ['sleep', 'infinity']
  - name: dockerstore
commands:
  - id: setupscript
    exec:
      component: test
      commandLine: "chmod +x script.sh"
      workingDir: /projects/devfiles
  - id: executescript
    exec:
      component: test
      commandLine: "/projects/devfiles/script.sh"
  - id: yumupdate
    exec:
      component: test
      commandLine: "yum -y update --security"
events:
  postStart:
    - setupscript
    - executescript
    - yumupdate
```

Los registros de inicio, comandos y eventos de Devfile se capturan y almacenan en `/aws/mde/Logs`. Para depurar el comportamiento de los archivos de desarrollo, inicie su entorno de desarrollo con un archivo de desarrollo que funcione y acceda a los registros.

Comandos de Devfile

Actualmente, CodeCatalyst solo admite `exec` comandos en su archivo de desarrollo. Para obtener más información, consulta [Cómo añadir comandos](#) en la documentación de DevFile.io.

El siguiente ejemplo muestra cómo especificar `exec` comandos en el devfile.

```
commands:
  - id: setupscript
    exec:
      component: test
```

```
    commandLine: "chmod +x script.sh"
    workingDir: /projects/devfiles
- id: executescript
  exec:
    component: test
    commandLine: "./projects/devfiles/script.sh"
- id: updateyum
  exec:
    component: test
    commandLine: "yum -y update --security"
```

Una vez conectado a su entorno de desarrollo, puede ejecutar los comandos definidos a través del terminal.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescript
```

En el caso de comandos de ejecución prolongada, puedes usar el indicador de streaming `-s` para mostrar la ejecución del comando en tiempo real.

```
/aws/mde/mde -s command <command-id>
```

Note

`command-id` debe estar en minúsculas.

Los parámetros de ejecución son compatibles con CodeCatalyst

CodeCatalyst admite los siguientes `exec` parámetros en la versión 2.0.0 de `devfile`.

- `commandLine`
- `component`
- `id`
- `workingDir`

Eventos de Devfile

Actualmente, CodeCatalyst solo admite `postStart` eventos en tu archivo de desarrollo. Para obtener más información, consulta la documentación [postStartObject](#) de DevFile.io.

En el siguiente ejemplo, se muestra cómo añadir enlaces de `postStart` eventos en el devfile.

```
commands:
  - id: executescript
    exec:
      component: test
      commandLine: "./projects/devfiles/script.sh"
  - id: updateyum
    exec:
      component: test
      commandLine: "yum -y update --security"
events:
  postStart:
    - updateyum
    - executescript
```

Tras el inicio, su entorno de desarrollo ejecutará los `postStart` comandos especificados en el orden en que estén definidos. Si un comando falla, el entorno de desarrollo seguirá ejecutándose y el resultado de la ejecución se almacenará en los registros de `abajo/aws/mde/logs`.

Componentes de Devfile

Actualmente, CodeCatalyst solo admite `container` los componentes de su archivo de desarrollo. Para obtener más información, consulta [Cómo añadir componentes](#) en la documentación de DevFile.io.

En el siguiente ejemplo, se muestra cómo añadir un comando de inicio al contenedor de su devfile.

```
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      command: ['sleep', 'infinity']
```

Note

Si el contenedor tiene un comando de entrada de corta duración, debes incluirlo `command: ['sleep', 'infinity']` para que el contenedor siga funcionando.

CodeCatalyst también admite las siguientes propiedades en el componente de contenedor: `argsenv`, `mountSources`, `yvolumeMounts`.

Imágenes universales devfile

La imagen universal predeterminada incluye los lenguajes de programación más utilizados y las herramientas relacionadas que se pueden usar para su IDE. Si no se especifica ninguna imagen, CodeCatalyst proporciona esta imagen y contiene herramientas mantenidas por CodeCatalyst. Para mantenerse informado de las nuevas versiones de imágenes, consulte [Notificaciones de imagen universales con SNS](#).

Note

La `public.ecr.aws/aws-mde/universal-image:latest` imagen obtiene la `public.ecr.aws/aws-mde/universal-image:3.0` imagen.

Amazon CodeCatalyst admite las siguientes imágenes de devfile.

Versión de imagen	Identificador de imagen
Universal image 1.0	<code>public.ecr.aws/aws-mde/universal-image:1.0</code>
Universal image 2.0	<code>public.ecr.aws/aws-mde/universal-image:2.0</code>
Universal image 3.0	<code>public.ecr.aws/aws-mde/universal-image:3.0</code>

Note

Si lo estás usando AWS Cloud9, la función de autocompletar no funcionará para PHP, Ruby y CSS después de la actualización `universal-image:3.0`.

Temas

- [Notificaciones de imagen universales con SNS](#)
- [Versiones en tiempo de ejecución de Universal Image 1.0](#)
- [Versiones en tiempo de ejecución de Universal Image 2.0](#)
- [Versiones en tiempo de ejecución de Universal Image 3.0](#)

Notificaciones de imagen universales con SNS

CodeCatalyst proporciona un servicio universal de notificación de imágenes. Puede utilizarla para suscribirse a un tema del Amazon Simple Notification Service (SNS) que le notifique cuando se publiquen actualizaciones de imágenes CodeCatalyst universales. Para obtener más información sobre los temas de SNS, consulte [¿Qué es Amazon Simple Notification Service?](#).

Cada vez que se publican nuevas imágenes universales, enviamos notificaciones a los suscriptores; en esta sección se describe cómo suscribirse a las actualizaciones de imágenes CodeCatalyst universales.

Ejemplo de mensaje

```
{
  "Type": "Notification",
  "MessageId": "123456789",
  "TopicArn": "arn:aws:sns:us-east-1:1234657890:universal-image-updates",
  "Subject": "New Universal Image Release",
  "Message": {
    "v1": {
      "Message": "A new version of the Universal Image has been released. You are now able to launch new DevEnvironments using this image.",
      "image": {
        "release_type": "MAJOR VERSION",
        "image_name": "universal-image",
        "image_version": "2.0",
```

```
        "image_uri": "public.ecr.aws/amazonlinux/universal-image:2.0"
    }
}
},
"Timestamp": "2021-09-03T19:05:57.882Z",
"UnsubscribeURL": "example url"
}
```

Para suscribirse a las actualizaciones CodeCatalyst universales de imágenes mediante la consola Amazon SNS

1. [Abra la consola de Amazon SNS en el panel de control.](#)
2. En la barra de navegación, elija su. Región de AWS
3. En el panel de navegación, seleccione Suscripciones y, a continuación, elija Crear suscripción.
4. En el ARN del tema, introduzca. `arn:aws:sns:us-east-1:089793673375:universal-image-updates`
5. En Protocol (Protocolo), elija Email (Correo electrónico).
6. En Endpoint, proporciona una dirección de correo electrónico. Esta dirección de correo electrónico se utilizará para recibir las notificaciones.
7. Seleccione Crear suscripción.
8. Recibirás un correo electrónico de confirmación con el asunto «AWS Notificación: confirmación de suscripción». Abre el correo electrónico y selecciona Confirmar la suscripción.

Para cancelar la suscripción a las actualizaciones CodeCatalyst universales de imágenes mediante la consola Amazon SNS

1. [Abra la consola de Amazon SNS en el panel de control.](#)
2. En la barra de navegación, elija su. Región de AWS
3. En el panel de navegación, selecciona Suscripciones y, a continuación, selecciona la suscripción de la que quieres darte de baja.
4. Selecciona Acciones y, a continuación, selecciona Eliminar suscripciones.
5. Elija Eliminar.

Versiones en tiempo de ejecución de Universal Image 1.0

En la siguiente tabla se enumeran los tiempos de ejecución disponibles para `universal-image:1.0`.

`universal-image:1.0` versiones en tiempo de ejecución

Nombre del entorno de tiempo de ejecución	Versión	Versión principal específica y última versión secundaria
aws cli	2.11	aws-cli: 2.x
docker compose	2.16	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x
golang	1.19	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	14,20	nodejs: 14.x
	16,19	nodejs: 16.x
openssl	1.0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3.10	
ruby	3.1	ruby: 3.x
terraform	1.4	terraform: 1.x

Versiones en tiempo de ejecución de Universal Image 2.0

En la siguiente tabla se enumeran los tiempos de ejecución disponibles para `universal-image:2.0`.

`universal-image:2.0` versiones en tiempo de ejecución

Nombre del entorno de tiempo de ejecución	Versión	Versión principal específica y última versión secundaria
aws cli	2.11	aws-cli: 2.x
docker compose	2.17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x
golang	1,20	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	16.19	nodejs: 16.x
openssl	1.0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3.10	
ruby	3.2	ruby: 3.x
terraform	1.4	terraform: 1.x

Versiones en tiempo de ejecución de Universal Image 3.0

En la siguiente tabla se enumeran los tiempos de ejecución disponibles para `universal-image:3.0`.

universal-image:3.0 versiones en tiempo de ejecución

Nombre del entorno de tiempo de ejecución	Versión	Versión principal específica y última versión secundaria
aws cli	2.11	aws-cli: 2.x
docker compose	2.17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x
golang	1.21	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	18.17	nodejs: 18.x
	20.6	nodejs: 20.x
openssl	3.0	openssl: 3.x
php	8.2	php: 8.x
python	3.9	python: 3.x
	3.11	
ruby	3.2	ruby: 3.x
terraform	1.5	terraform: 1.x

Uso de entornos de desarrollo con una conexión de VPC

Una conexión de VPC es un CodeCatalyst recurso que contiene todas las configuraciones necesarias para que su flujo de trabajo acceda a una VPC. Los administradores del espacio pueden añadir sus propias conexiones de VPC en la CodeCatalyst consola de Amazon en nombre de los miembros del espacio. Al agregar una conexión de VPC, los miembros del espacio pueden ejecutar acciones de flujo de trabajo y crear entornos de desarrollo que cumplan con las reglas de la red y puedan acceder a los recursos de la VPC asociada.

Solo puede asociar un entorno de desarrollo a una conexión de VPC al crear un entorno de desarrollo. No puedes cambiar la conexión de VPC asociada a tu entorno de desarrollo después de crearlo. Si quieres usar una conexión de VPC diferente, tienes que eliminar tu entorno de desarrollo actual y crear uno nuevo.

Important

Los entornos de desarrollo con una conexión de VPC no admiten [repositorios de fuentes de terceros vinculados](#) a ellos. CodeCatalyst

Tenga en cuenta que los entornos de desarrollo utilizan varios AWS recursos y servicios al crearse. Esto significa que los entornos de desarrollo se conectan a los siguientes AWS servicios:

- Amazon CodeCatalyst
- AWS SSM
- AWS KMS
- Amazon ECR
- Amazon CloudWatch
- Amazon ECS

Note

AWS Toolkit no admite la creación de entornos de desarrollo con una conexión de VPC asociada. Ten en cuenta también que si utilizas un IDE distinto de este AWS Cloud9, es posible que los tiempos de carga sean de unos cinco minutos.

Debe tener el rol de administrador de espacio o el rol de usuario avanzado para administrar las conexiones de VPC a nivel de espacio. Para obtener más información sobre las VPC, consulte [Gestión de las VPC de Amazon CodeCatalyst en](#) la Guía del CodeCatalyst administrador.

Uso de entornos de desarrollo con un IDE

Puede usar Dev Environments para trabajar rápidamente en el código almacenado en los repositorios de código fuente de su proyecto. Los entornos de desarrollo aumentan la velocidad de desarrollo porque puede empezar a programar inmediatamente en un entorno de desarrollo en la nube totalmente funcional y específico para cada proyecto, con un entorno de desarrollo integrado (IDE) compatible.

Para obtener información sobre cómo trabajar CodeCatalyst desde un IDE, consulte la siguiente documentación.

- [Amazon CodeCatalyst para JetBrains IDE](#)
- [Amazon CodeCatalyst para VS Code](#)
- [Amazon CodeCatalyst para AWS Cloud9](#)

Cuotas para entornos de desarrollo en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de los entornos de desarrollo en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Número de horas de entorno de desarrollo al mes	Las horas del entorno de desarrollo se ven afectadas por los límites generales de almacenamiento de su espacio. Para obtener más información, consulte Precios de y Solución de problemas con entornos de desarrollo .
Cantidad de almacenamiento en el entorno de desarrollo por espacio	El almacenamiento del entorno de desarrollo o se ve afectado por los límites generales de almacenamiento de tu espacio. Para obtener más información, consulte Precios

	de y Solución de problemas con entornos de desarrollo .
Cantidad de cómputo en el entorno de desarrollo	El cómputo del entorno de desarrollo se ve afectado por los límites generales de almacenamiento de su espacio. Para obtener más información, consulte Precios de y Solución de problemas con entornos de desarrollo .

Paquetes en CodeCatalyst

Amazon CodeCatalyst incluye un servicio de repositorio de paquetes totalmente gestionado que permite a su equipo de desarrollo almacenar y compartir de forma segura los paquetes de software utilizados para el desarrollo de aplicaciones. Estos paquetes se almacenan en repositorios de paquetes, que se crean y organizan dentro de los CodeCatalyst proyectos.

CodeCatalyst admite los siguientes formatos de paquetes:

- npm

Los paquetes de un repositorio de paquetes se pueden detectar y compartir entre los miembros del proyecto que contiene el repositorio.

Para añadir paquetes a un repositorio, configure un administrador de paquetes para que utilice el punto final (URL) del repositorio. A continuación, puede usar el administrador de paquetes para publicar paquetes en el repositorio.

Puede configurar los CodeCatalyst flujos de trabajo para publicar paquetes en los repositorios de paquetes y consumirlos desde ellos. CodeCatalyst Para obtener más información sobre el uso de paquetes en los flujos de trabajo, consulte [Trabajar con paquetes](#).

Puede hacer que los paquetes de un repositorio de paquetes estén disponibles en otro repositorio del mismo proyecto agregándolo como repositorio principal. Todas las versiones de paquetes disponibles en el repositorio ascendente también están disponibles en el repositorio descendente.

Puedes hacer que los paquetes de código abierto estén disponibles en tu CodeCatalyst repositorio conectando repositorios públicos y externos a él. Para obtener más información sobre los repositorios originales y la conexión a repositorios externos, incluida una lista de los repositorios compatibles, consulte. [Trabajar con repositorios ascendentes](#)

Temas

- [Conceptos de paquetes](#)
- [Trabajar con repositorios de paquetes](#)
- [Trabajar con repositorios ascendentes](#)
- [Conexión a repositorios públicos externos](#)
- [Trabajar con paquetes](#)

- [Con npm](#)
- [Cuotas para paquetes](#)

Conceptos de paquetes

Estos son algunos conceptos y términos que debes conocer a la hora de gestionar, publicar o consumir paquetes CodeCatalyst.

Paquetes

Un paquete es un paquete que incluye el software y los metadatos necesarios para instalar el software y resolver cualquier dependencia. CodeCatalyst admite el formato de paquete npm.

Un paquete se compone de:

- Un nombre (por ejemplo, webpack es el nombre de un paquete npm popular)
- Un espacio de [nombres](#) opcional (por ejemplo, en) @types @types/node
- Un conjunto de [versiones](#) (por ejemplo,,1.0.0,1.0.1) 1.0.2
- Metadatos a nivel de paquete (por ejemplo, etiquetas npm dist)

Espacios de nombres de paquetes

Algunos formatos de paquetes admiten nombres de paquetes jerárquicos para organizar los paquetes en grupos lógicos y evitar colisiones de nombres. Los paquetes que tienen el mismo nombre se pueden almacenar en diferentes espacios de nombres. Por ejemplo, npm admite ámbitos y el paquete npm @types/node tiene un ámbito y un nombre de @types. node Hay muchos otros nombres de paquetes en el alcance @types. En CodeCatalyst, el ámbito («tipos») se denomina espacio de nombres del paquete y el nombre («nodo») se denomina nombre del paquete. Si no tienes una forma de agrupar los nombres de los paquetes, puede resultar más difícil evitar las colisiones de nombres.

Versiones de paquetes

La versión de un paquete identifica la versión específica de un paquete, por ejemplo @types/node@12.6.9. El formato y la semántica del número de versión varían según los distintos formatos de paquete. Por ejemplo, las versiones del paquete npm deben cumplir con la [especificación de](#)

[control de versiones semántico](#). En CodeCatalyst, una versión de paquete consta del identificador de la versión, `package-version-level` los metadatos y un conjunto de activos.

Activos

Un activo es un archivo individual almacenado en el CodeCatalyst que se asocia a una versión del paquete, como un `.tgz` archivo npm.

Repositorios de paquetes

Un repositorio de CodeCatalyst paquetes contiene un conjunto de [paquetes](#), que contienen [versiones de paquetes](#), cada una de las cuales se asigna a un conjunto de [activos](#). Cada repositorio de paquetes proporciona puntos finales para obtener y publicar paquetes mediante herramientas como la CLI (`npm`) de Node.js. Puede crear hasta 1000 repositorios de paquetes en cada espacio.

Puede vincular un repositorio de paquetes a otro repositorio mediante los repositorios anteriores. Al vincular un repositorio de paquetes como repositorio ascendente, puede usar los paquetes del repositorio vinculado a través del repositorio configurado. Para obtener más información, consulte [Repositorios upstream](#).

Los repositorios Gateway son un tipo especial de repositorio de paquetes que extrae y almacena paquetes de las autoridades oficiales de empaquetado externas. Para obtener más información, consulte [Los repositorios Gateway](#).

Los repositorios Gateway

Un repositorio de puerta de enlace es un tipo especial de repositorio de paquetes que está conectado a una autoridad de paquetes oficial y externa compatible. Al añadir un repositorio de puerta de enlace como [repositorio principal](#), puede consumir paquetes de la autoridad de empaquetado oficial correspondiente. Tu repositorio descendente no se comunica con el repositorio público, sino que todo está intermediado por el repositorio gateway. Los paquetes que se consumen de esta manera se almacenan tanto en el repositorio de puerta de enlace como en el repositorio descendente que recibió la solicitud original.

Los repositorios Gateway están predefinidos, pero deben crearse en cada proyecto que se vaya a utilizar. La siguiente lista contiene todos los repositorios de puertas de enlace en los que se pueden crear CodeCatalyst y la autoridad de paquetes a la que están conectados.

- `npm-public-registry-gateway` proporciona paquetes npm de `npmjs.com`.

Repositorios upstream

Puede utilizarlos CodeCatalyst para crear una relación ascendente entre dos repositorios de paquetes. Un repositorio de paquetes es un repositorio anterior a otro cuando se puede acceder a las versiones de paquetes que contiene desde el punto final del repositorio de paquetes del repositorio descendente. Con una relación ascendente, el contenido de los dos repositorios de paquetes se fusiona de forma efectiva desde el punto de vista del cliente.

Por ejemplo, si un administrador de paquetes solicita una versión de paquete que no existe en un repositorio, CodeCatalyst buscará la versión del paquete en los repositorios ascendentes configurados. Los repositorios ascendentes se buscan en el orden en que están configurados y, una vez que se encuentra un paquete, CodeCatalyst se detiene la búsqueda.

Trabajar con repositorios de paquetes

En CodeCatalyst, los paquetes se almacenan y administran en repositorios de paquetes. Para publicar paquetes en un repositorio de paquetes CodeCatalyst o consumirlos de uno CodeCatalyst (o de cualquier repositorio de paquetes público compatible), debe crear un repositorio de paquetes y conectar su administrador de paquetes a él.

Temas

- [Crear un repositorio de paquetes](#)
- [Conectarse a un repositorio de paquetes](#)
- [Edición de un repositorio de paquetes](#)
- [Eliminar un repositorio de paquetes](#)

Crear un repositorio de paquetes

Realice los siguientes pasos para crear un repositorio de paquetes en CodeCatalyst.

Para crear un repositorio de paquetes

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee crear un repositorio de paquetes.
3. En el panel de navegación, elija Paquetes.
4. En la página Repositorios de paquetes, elija Crear repositorio de paquetes.

5. En la sección de detalles del repositorio de Package, agrega lo siguiente:
 - a. Nombre del repositorio. Considere la posibilidad de usar un nombre descriptivo con detalles como el nombre de su proyecto o equipo, o cómo se usará el repositorio.
 - b. (Opcional) Descripción del repositorio. La descripción del repositorio es especialmente útil cuando tienes varios repositorios en varios equipos de un proyecto.
6. En la sección Editar repositorios originales, añade los repositorios de paquetes a los que quieras acceder a través de tu repositorio de paquetes. CodeCatalyst Puedes añadir repositorios de Gateway para conectarte a repositorios de paquetes externos o a otros repositorios de paquetes. CodeCatalyst
 - Cuando se solicita un paquete desde un repositorio de paquetes, se buscarán en los repositorios anteriores en el orden en que aparecen en esta lista. Una vez que se encuentre un paquete, se CodeCatalyst detendrá la búsqueda. Para cambiar el orden de los repositorios originales, puede arrastrar y soltar los repositorios en la lista o utilizar los botones Reordenar.
7. Elija Crear para crear su repositorio de paquetes.

Conectarse a un repositorio de paquetes

Para publicar paquetes en CodeCatalyst o desde los que consumirlos CodeCatalyst, debe configurar el administrador de paquetes con la información y las CodeCatalyst credenciales del punto final del repositorio de paquetes. Si no ha creado un repositorio, puede hacerlo siguiendo las instrucciones que se indican en [Crear un repositorio de paquetes](#).

Para obtener instrucciones sobre cómo conectar el administrador de paquetes npm a un repositorio de CodeCatalyst paquetes, consulte [Configuración y uso de npm](#).

Edición de un repositorio de paquetes

Realice los siguientes pasos para editar la descripción de un repositorio de paquetes y los repositorios anteriores.

Para editar un repositorio de paquetes

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto que contiene el repositorio de paquetes que desea editar.

3. En el panel de navegación, elija Paquetes.
4. En la página Package repositories, elija el repositorio que desee eliminar.
5. Seleccione el menú desplegable Acciones y seleccione Editar.
6. Edite la descripción del repositorio y los repositorios originales. Para obtener más información sobre los repositorios ascendentes, consulte [Trabajar con repositorios ascendentes](#).
7. Seleccione Guardar.

Eliminar un repositorio de paquetes

Realice los siguientes pasos para eliminar un repositorio de paquetes en CodeCatalyst.

Para eliminar un repositorio de paquetes

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto que contiene el repositorio de paquetes que desea eliminar.
3. En el panel de navegación, elija Paquetes.
4. En la página Package repositories, elija el repositorio que desee eliminar.
5. Seleccione el menú desplegable Acciones y seleccione Eliminar.
6. Revise la información proporcionada sobre los efectos de eliminar un repositorio de paquetes.
7. Entre de1ete en el campo de entrada y elija Eliminar.

Trabajar con repositorios ascendentes

Puede conectar tanto los repositorios de puerta de enlace como otros repositorios de paquetes como archivos ascendentes a sus repositorios de paquetes. Esto permite a un cliente administrador de paquetes acceder a los paquetes que se encuentran en más de un repositorio de paquetes mediante un único punto final del repositorio de paquetes. Las principales ventajas de utilizar los repositorios originales son las siguientes:

- Solo tienes que configurar tu administrador de paquetes con un único punto final de repositorio para extraerlos de múltiples fuentes.
- Los paquetes consumidos de un repositorio principal se almacenan en el repositorio descendente, lo que garantiza que los paquetes estén disponibles incluso si el repositorio principal sufre interrupciones inesperadas.

Puedes añadir repositorios ascendentes al crear un repositorio de paquetes. También puedes añadir o eliminar repositorios ascendentes de los repositorios de paquetes existentes en la consola. CodeCatalyst

Al añadir un repositorio de puerta de enlace como repositorio ascendente, el repositorio de paquetes se conecta al repositorio de paquetes público correspondiente del repositorio de puerta de enlace. Para obtener una lista de los repositorios de paquetes públicos compatibles, consulte. [Repositorios de paquetes externos compatibles y sus repositorios de puerta de enlace](#)

Puede vincular varios repositorios como repositorios ascendentes. Por ejemplo, supongamos que tu equipo crea un repositorio con el nombre `project-repo` y ya está utilizando otro repositorio con el nombre `team-repo` `npm-public-registry-gateway` agregado como repositorio ascendente, que está conectado al repositorio público de `npm.npmjs.com`. Puedes añadirlo `team-repo` como repositorio ascendente a `project-repo`. En este caso, solo tiene que configurar su administrador de paquetes para que lo utilice `project-repo` para extraer paquetes de `project-repo`, `team-repo` `npm-public-registry-gateway`, `npmjs.com`.

Temas

- [Añadir un repositorio ascendente](#)
- [Edición del orden de búsqueda de los repositorios ascendentes](#)
- [Solicitar una versión de paquete con repositorios ascendentes](#)
- [Eliminar un repositorio anterior](#)

Añadir un repositorio ascendente

Al agregar un repositorio de paquetes público u otro CodeCatalyst repositorio de paquetes como repositorio ascendente a su repositorio descendente, todos los paquetes del repositorio ascendente estarán disponibles para los administradores de paquetes que estén conectados al repositorio descendente.

Para añadir un repositorio ascendente

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, elija el repositorio de paquetes al que desee añadir un repositorio ascendente.
3. Selecciona el menú desplegable Acciones y selecciona Editar.
4. En la sección Repositorios ascendentes, selecciona Añadir repositorio ascendente.

5. Selecciona la barra de búsqueda para que aparezca una lista de los repositorios disponibles y busca. Puedes añadir repositorios de paquetes públicos compatibles u otros repositorios como CodeCatalyst repositorios ascendentes. Cuando encuentres el repositorio que quieres añadir, selecciónalo de la lista.

 Note

En los repositorios de Gateway, hay un npm-public-registry-gatewayrepositorio. Para conectarse a un servidor público externo, las autoridades encargadas de empaquetar paquetes, como npmjs.com, CodeCatalyst utilizan un repositorio de puerta de enlace como repositorios intermediarios que buscan y almacenan los paquetes extraídos de repositorios externos. Esto ahorra tiempo y transferencia de datos, ya que todos los repositorios de paquetes de un proyecto utilizarán paquetes del repositorio de puerta de enlace.

6. Cuando hayas seleccionado todos los repositorios que quieres añadir como repositorios ascendentes, selecciona Añadir.
7. Para obtener más información sobre cómo cambiar el orden de búsqueda de los repositorios ascendentes, consulte [Edición del orden de búsqueda de los repositorios ascendentes](#)

Cuando hayas agregado un repositorio principal, puedes usar un administrador de paquetes que esté conectado a tu repositorio local para recuperar los paquetes del repositorio ascendente. No necesitas actualizar la configuración del administrador de paquetes. Para obtener más información sobre cómo solicitar versiones de paquetes desde un repositorio principal, consulte [Solicitar una versión de paquete con repositorios ascendentes](#).

Edición del orden de búsqueda de los repositorios ascendentes

CodeCatalyst busca en los repositorios ascendentes en el orden de búsqueda configurado. Cuando se encuentra un paquete, CodeCatalyst deja de buscar. Puede cambiar el orden en el que se buscan los paquetes en los repositorios originales.

Para editar el orden de búsqueda de los repositorios ascendentes

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios, elija el repositorio de paquetes cuyo orden de búsqueda en los repositorios ascendentes desee editar.

3. Selecciona el menú desplegable Acciones y selecciona Editar.
4. En la sección Repositorios ascendentes, puedes ver los repositorios ascendentes y su orden de búsqueda. Para cambiar el orden de búsqueda, arrastre y suelte los repositorios en la lista o utilice los botones Reordenar.
5. Cuando termines de editar el orden de búsqueda de los repositorios originales, selecciona Guardar.

Solicitar una versión de paquete con repositorios ascendentes

El siguiente ejemplo muestra los posibles escenarios en los que un administrador de paquetes solicita un paquete de un repositorio de CodeCatalyst paquetes que tiene repositorios ascendentes.

En este ejemplo, un administrador de paquetes, por ejemplo `plonpm`, solicita una versión de paquete desde un repositorio de paquetes denominado `downstream` que tiene varios repositorios ascendentes. Cuando se solicita el paquete, puede ocurrir lo siguiente:

- Si `downstream` contiene la versión del paquete solicitada, se devuelve al cliente.
- Si `downstream` no contiene la versión del paquete solicitada, la CodeCatalyst busca en `downstream` sus repositorios originales, en el orden de búsqueda configurado. Si se encuentra la versión del paquete, se copia una referencia a la misma en `downstream` y la versión del paquete se devuelve al cliente.
- Si ninguno `downstream` de sus repositorios principales contiene la versión del paquete, se devuelve una `Not Found` respuesta HTTP 404 al cliente.

El número máximo de repositorios ascendentes directos permitidos para un repositorio es 10. El número máximo de CodeCatalyst búsquedas en los repositorios cuando se solicita una versión de paquete es de 25.

Retención de paquetes de repositorios ascendentes

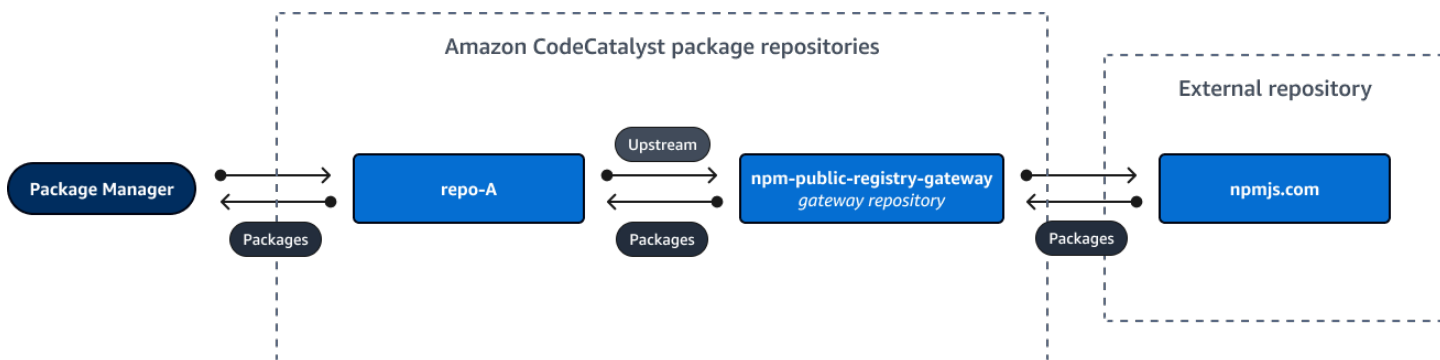
Si la versión de un paquete solicitada se encuentra en un repositorio anterior, se conserva una referencia a la misma y siempre está disponible en el repositorio que la solicitó. Esto garantiza que tendrá acceso a sus paquetes en caso de que se produzca una interrupción inesperada del repositorio principal. La versión del paquete retenida no se ve afectada por ninguno de los siguientes factores:

- Eliminar el repositorio ascendente.

- Desconectar el repositorio ascendente del repositorio descendente.
- Eliminar la versión del paquete del repositorio ascendente.
- Editar la versión del paquete en el repositorio ascendente (por ejemplo, añadiéndole un nuevo activo).

Obtener paquetes a través de una relación ascendente

CodeCatalyst puede buscar paquetes a través de varios repositorios enlazados denominados repositorios ascendentes. Si un repositorio de CodeCatalyst paquetes tiene una conexión ascendente a otro repositorio de CodeCatalyst paquetes que tiene una conexión ascendente a un repositorio de puerta de enlace, las solicitudes de paquetes que no estén en el repositorio ascendente se copian del repositorio externo. Por ejemplo, considere la siguiente configuración: un repositorio denominado `repo-A` tiene una conexión ascendente con el repositorio de puerta de enlace, `npm-public-registry-gateway` [tiene una conexión ascendente al repositorio público de paquetes, https://npmjs.com](https://npmjs.com).



Si npm está configurado para usar el `repo-A` repositorio, la ejecución `npm install` inicia la copia de los paquetes de <https://npmjs.com> a `npm-public-registry-gateway`. También se incluyen las versiones instaladas en `repo-A`. El siguiente ejemplo instala `lodash`.

```
$ npm config get registry
https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/
$ npm install lodash
+ lodash@4.17.20
added 1 package from 2 contributors in 6.933s
```

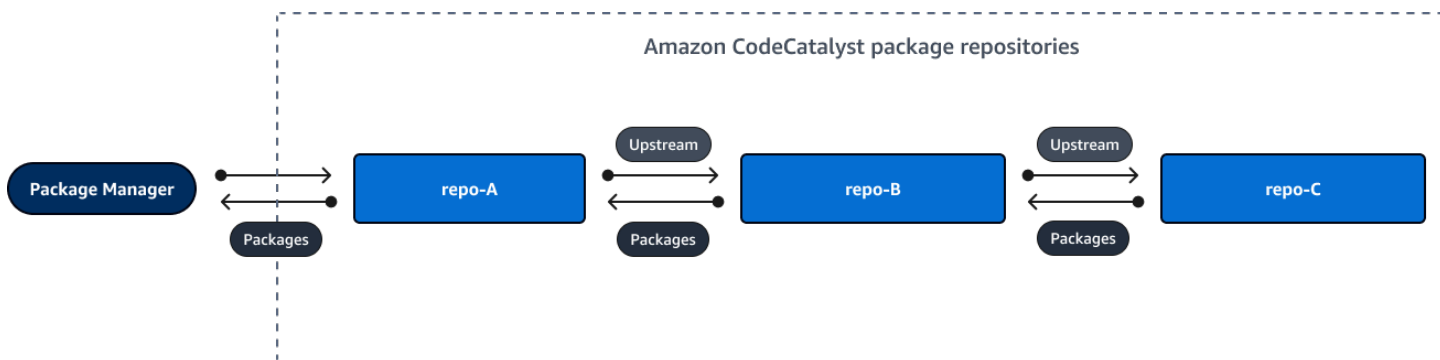
Tras ejecutarse `npm install`, `repo-A` contiene solo la versión más reciente (`lodash 4.17.20`), ya que es la versión desde la que se obtuvo. `npm repo-A`

Como `npm-public-registry-gateway` tiene una conexión ascendente externa a <https://npmjs.com>, todas las versiones de los paquetes que se importan de <https://npmjs.com> se almacenan en ella. `npm-public-registry-gateway` Estas versiones de paquetes podrían haber sido recuperadas por cualquier repositorio descendente con una conexión ascendente a `npm-public-registry-gateway`.

El contenido de `npm-public-registry-gateway` proporciona una forma de ver todos los paquetes y las versiones de los paquetes importados desde <https://npmjs.com> a lo largo del tiempo.

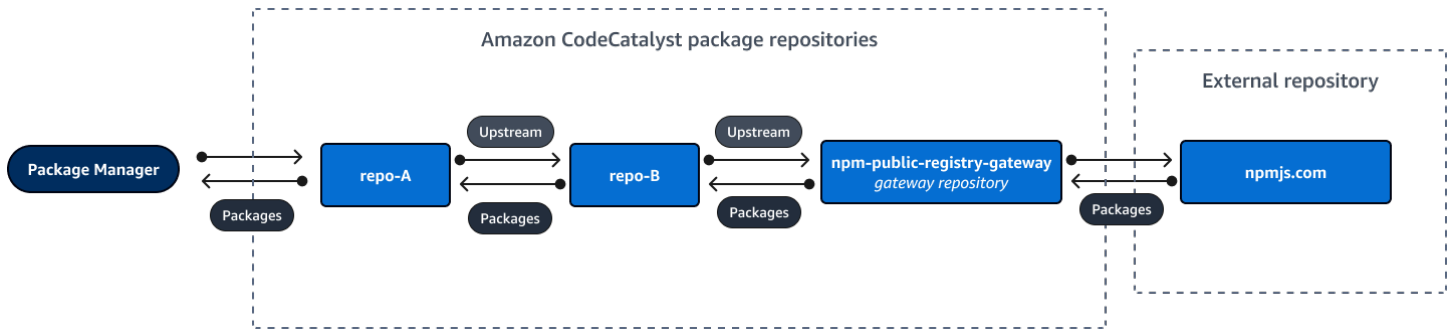
Retención de paquetes en repositorios intermedios

CodeCatalyst permite encadenar repositorios ascendentes. Por ejemplo, `repo-A` puede tener `repo-B` como repositorio ascendente y `repo-B` puede tenerlo `repo-C` como repositorio ascendente. Esta configuración hace que las versiones del paquete en `repo-B` y `repo-C` estén disponibles de `repo-A`.



Cuando un administrador de paquetes se conecta al repositorio `repo-A` y obtiene una versión del paquete del repositorio `repo-C`, la versión del paquete no se conserva en el repositorio `repo-B`. La versión del paquete solo se conserva en el repositorio descendente más alejado, como ocurre en este ejemplo. `repo-A` No se conserva en ningún repositorio intermedio. Esto también es válido para cadenas más largas; por ejemplo, si hubiera cuatro repositorios: `repo-A`, `repo-B`, `repo-C` y `repo-D`, y un administrador de paquetes conectado para `repo-A` extraer una versión de un paquete `repo-D`, la versión del paquete se conservaría en `repo-A` pero no en `repo-B` o `repo-C`.

El comportamiento de retención de paquetes es similar cuando se extrae una versión de paquete de un repositorio de paquetes público, excepto que la versión del paquete siempre se conserva en el repositorio de puerta de enlace que tiene la conexión ascendente directa con el repositorio público. Por ejemplo, `repo-A` tiene `repo-B` como repositorio ascendente. `repo-B` tiene `npm-public-registry-gateway` como repositorio ascendente, que tiene una conexión ascendente con el repositorio público `npmjs.com`; consulte el siguiente diagrama.



Si un administrador de paquetes conectado **repo-A** solicita una versión de paquete específica, por ejemplo, lodash 4.17.20, y la versión del paquete no está presente en ninguno de los tres repositorios, se obtendrá de npmjs.com. Cuando se obtiene lodash 4.17.20, se conserva porque es el repositorio descendente más alejado y tiene la conexión ascendente con el repositorio externo público, npmjs.com. **repo-A** **npm-public-registry-gateway** lodash 4.17.20 no se conserva porque se trata de un repositorio intermedio. **repo-B**

Eliminar un repositorio anterior

Si ya no desea acceder a los paquetes de un repositorio ascendente, puede eliminar el repositorio ascendente de un repositorio de paquetes.

Warning

Al eliminar un repositorio anterior, podrías romper las cadenas de relaciones ascendentes, lo que podría interrumpir tus proyectos o compilaciones.

Para eliminar un repositorio anterior

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, elija el repositorio de paquetes del que quiere eliminar un repositorio anterior.
3. Selecciona el menú desplegable Acciones y selecciona Editar.
4. En la sección Repositorios ascendentes, busca el repositorio ascendente que deseas eliminar y selecciona Eliminar.
5. Cuando hayas terminado de eliminar los repositorios anteriores, selecciona Guardar.

Conexión a repositorios públicos externos

Para conectar los repositorios de CodeCatalyst paquetes a los repositorios externos públicos compatibles, añada el repositorio de puerta de enlace correspondiente como repositorio ascendente. Los repositorios Gateway actúan como repositorios intermediarios que buscan y almacenan paquetes extraídos de repositorios externos. Esto ahorra tiempo y transferencia de datos, ya que todos los repositorios de paquetes de un proyecto utilizan paquetes del repositorio de puerta de enlace.

Para conectarse a un repositorio público mediante repositorios de puerta de enlace

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, en los repositorios de Gateway, puede ver una lista de los repositorios de gateway compatibles y sus descripciones. Para usar un repositorio de puerta de enlace, primero debe crearlo. Si se ha creado el repositorio de puerta de enlace, se muestran la fecha y la hora en que se creó. Si no lo ha hecho, elija Crear para crearlo.
3. Elija el repositorio de paquetes que desee conectar al repositorio público.
4. Selecciona el menú desplegable Acciones y selecciona Editar.
5. Para conectarte al repositorio público, agrega el repositorio de puerta de enlace que corresponda al repositorio público al que deseas conectarte como repositorio principal.

En la sección Editar repositorios ascendentes, selecciona Añadir repositorio. CodeCatalyst

6. La sección Repositorios de Gateway muestra una lista de todos los repositorios de gateway disponibles. Cuando encuentres el repositorio de puerta de enlace que corresponda al repositorio público y externo al que deseas conectarte, selecciónalo de la lista y selecciona Agregar.
7. Cuando se solicita un paquete desde un repositorio, CodeCatalyst busca en los repositorios ascendentes en el orden en que aparecen en la lista Editar repositorios ascendentes. Cuando se encuentra un paquete, deja de buscarlo. CodeCatalyst Para cambiar el orden de los repositorios anteriores, arrastre y suelte los repositorios en la lista o utilice las flechas de reordenación.
8. Cuando hayas terminado de añadir y ordenar los repositorios ascendentes, selecciona Guardar.

Cuando hayas agregado un repositorio de puerta de enlace como repositorio principal, puedes usar un administrador de paquetes que esté conectado a tu repositorio local para buscar paquetes del repositorio de paquetes público y externo que le corresponda. No necesitas actualizar la configuración del administrador de paquetes. Los paquetes que se consumen de esta manera se

almacenan tanto en el repositorio de la puerta de enlace como en el repositorio de paquetes local. Para obtener más información sobre cómo solicitar versiones de paquetes desde un repositorio principal, consulte [Solicitar una versión de paquete con repositorios ascendentes](#).

Repositorios de paquetes externos compatibles y sus repositorios de puerta de enlace

CodeCatalyst admite la adición de una conexión ascendente a las siguientes autoridades oficiales de paquetes con repositorios de puertas de enlace.

Tipo de paquete de repositorio	Descripción	Nombre del repositorio de Gateway
npm	registro público de npm	npm-public-registr y-gateway

Trabajar con paquetes

Un paquete CodeCatalyst es un paquete de software y los metadatos necesarios para resolver las dependencias e instalar el software. CodeCatalyst admite el formato de paquete npm. En esta sección se proporciona información sobre la publicación, la visualización y la eliminación de paquetes, así como sobre la actualización del estado de una versión del paquete.

Temas

- [Publicación de paquetes](#)
- [Ver los detalles de la versión del paquete](#)
- [Eliminar una versión de paquete](#)
- [Actualización del estado de la versión de un paquete](#)
- [Edición de los controles de origen del paquete](#)

Publicación de paquetes

Puede publicar versiones de cualquier tipo de paquete compatible en un repositorio de CodeCatalyst paquetes mediante las herramientas del administrador de paquetes.

Para obtener información sobre cómo conectar un administrador de paquetes a un CodeCatalyst repositorio de paquetes para publicar paquetes, consulte [Conectarse a un repositorio de paquetes](#).

Contenido

- [Repositorios editoriales y originales](#)
- [Paquetes privados y repositorios públicos](#)
- [Sobrescribir los activos del paquete](#)

Repositorios editoriales y originales

En CodeCatalyst, no puede publicar versiones de paquetes que estén presentes en repositorios ascendentes o repositorios públicos accesibles. Por ejemplo, supongamos que desea publicar un paquete npm en un repositorio de paquetes y que `myrepo` está conectado a `npmjs.com` a través de un repositorio gateway configurado como repositorio ascendente. `lodash@1.0` `myrepo` Si `lodash@1.0` está presente en el repositorio principal o en `npmjs.com`, CodeCatalyst rechaza cualquier intento de publicar en él emitiendo un error 409 de conflicto. `myrepo` Esto ayuda a evitar que publique accidentalmente un paquete con el mismo nombre y versión que un paquete en un repositorio principal, lo que puede provocar un comportamiento inesperado.

Puede seguir publicando diferentes versiones del nombre de un paquete que existan en un repositorio principal. Por ejemplo, si `lodash@1.0` está presente en un repositorio ascendente, pero no lo `lodash@1.1` está, puede `lodash@1.1` publicarlo en el repositorio descendente.

Paquetes privados y repositorios públicos

CodeCatalyst no publica los paquetes almacenados en los repositorios en CodeCatalyst repositorios públicos, como `npmjs.com`. CodeCatalyst importa paquetes de repositorios públicos a un CodeCatalyst repositorio, pero no mueve los paquetes en la dirección opuesta. Los paquetes que publiques en los CodeCatalyst repositorios permanecen privados y solo están disponibles para el CodeCatalyst proyecto al que pertenece el repositorio.

Sobrescribir los activos del paquete

No puedes volver a publicar un activo de paquete que ya existe con contenido diferente. Como npm solo admite un recurso por versión de paquete, para modificar una versión de paquete publicada, primero debes eliminarla.

Ver los detalles de la versión del paquete

Puede usar la CodeCatalyst consola para ver los detalles de una versión de paquete específica.

Para ver los detalles de la versión del paquete

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, elija el repositorio que contiene la versión del paquete de la que desea ver los detalles.
3. Busque la versión del paquete en la tabla Paquetes. Puede utilizar la barra de búsqueda para filtrar los paquetes por nombre de paquete. Elija el paquete de la lista.
4. En la página de detalles del paquete, elija Versiones y, a continuación, elija la versión que desee ver.

Eliminar una versión de paquete

Puede eliminar una versión del paquete desde la página de detalles de la versión del paquete de la CodeCatalyst consola.

Para eliminar una versión del paquete

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, elija el repositorio que contiene la versión del paquete que desea eliminar.
3. Busque y elija el paquete de la tabla.
4. En la página de detalles del paquete, selecciona Versiones y elige la versión que desees eliminar.
5. En la página de detalles de la versión del paquete, elija Acciones de la versión y, a continuación, elija Eliminar.
6. Escribe eliminar en el campo de texto y selecciona Eliminar.

Actualización del estado de la versión de un paquete

Cada versión del paquete CodeCatalyst tiene un estado que describe el estado actual y la disponibilidad de la versión del paquete. Puede cambiar el estado de la versión del paquete en la

CodeCatalyst consola. Para obtener más información sobre los posibles valores de estado de las versiones de los paquetes y sus significados, consulte [El estado de la versión del paquete](#).

Para actualizar el estado de una versión de paquete

1. En el panel de navegación, elija Packages (Paquetes).
2. En la página Repositorios de paquetes, elija el repositorio que contiene la versión del paquete cuyo estado desea actualizar.
3. Busque y elija el paquete de la tabla.
4. En la página de detalles del paquete, elija Versiones y, a continuación, elija la versión que desee ver.
5. En la página de detalles de la versión del paquete, elija Acciones y, a continuación, elija Eliminar de la lista, Archivar o Eliminar. Para obtener información sobre el estado de cada versión del paquete, consulte [El estado de la versión del paquete](#).
6. Introduzca el texto de confirmación en el campo de texto y, a continuación, seleccione Eliminar de la lista, Archivar o Eliminar, según el estado al que vaya a realizar la actualización.

El estado de la versión del paquete

Los siguientes son valores posibles para el estado de la versión del paquete. Puede cambiar el estado de la versión del paquete en la consola. Para obtener más información, consulte [Actualización del estado de la versión de un paquete](#).

- **Publicada:** la versión del paquete se publicó correctamente y un administrador de paquetes puede solicitarla. La versión del paquete se incluirá en las listas de versiones de paquetes devueltas a los administradores de paquetes; por ejemplo, en el resultado de `npm view <package-name> versions`. Todos los activos de la versión del paquete están disponibles en el repositorio.
- **Ocultada:** los activos de la versión del paquete están disponibles para su descarga desde el repositorio, pero la versión del paquete no se incluye en la lista de versiones devueltas a los administradores de paquetes. Por ejemplo, en el caso de un paquete npm, la salida de `npm view <package-name> versions` no incluye la versión del paquete. Esto significa que la lógica de resolución de dependencias de npm no selecciona la versión del paquete porque la versión no aparece en la lista de versiones disponibles. Sin embargo, si ya se hace referencia a la versión del paquete no listada en un `npm package-lock.json` archivo, aún se puede descargar e instalar; por ejemplo, cuando se está ejecutando `npm ci`.

- **Archivado:** los activos de la versión del paquete no se pueden descargar. La versión del paquete no se incluirá en la lista de versiones devueltas a los administradores de paquetes. Como los activos no están disponibles, los clientes bloquean el consumo de la versión del paquete. Si la compilación de la aplicación depende de una versión actualizada a Archived, la compilación fallará, a menos que la versión del paquete se haya almacenado en caché local. No puedes usar un administrador de paquetes o una herramienta de compilación para volver a publicar una versión de un paquete archivado porque todavía está presente en el repositorio. Sin embargo, puedes cambiar el estado de la versión del paquete a No publicada o Publicada en la consola.
- **Eliminado:** la versión del paquete no aparece en las listas y los activos no se pueden descargar del repositorio. La diferencia clave entre Eliminado y Archivado es que, con el estado Dispuesto, los activos de la versión empaquetada CodeCatalyst se eliminan de forma permanente. Por este motivo, no puede mover una versión de paquete de Eliminada a Archivada, No listada o Publicada. La versión del paquete no se puede utilizar porque se han eliminado los activos. Cuando una versión del paquete se marca como Eliminada, no se le facturará el almacenamiento de los activos del paquete.

Además de los estados de la lista anterior, también se puede eliminar una versión del paquete. Una vez eliminada, la versión del paquete no está en el repositorio y puede volver a publicarla libremente mediante un administrador de paquetes o una herramienta de compilación.

Edición de los controles de origen del paquete

En Amazon CodeCatalyst, las versiones de los paquetes se pueden añadir a un repositorio de paquetes publicándolas directamente, retirándolas de un repositorio anterior o incorporándolas desde un repositorio público externo. Si permites que las versiones de un paquete se añadan publicándolas directamente o ingiriéndolas desde repositorios públicos, eres vulnerable a un ataque de sustitución de dependencias. Para obtener más información, consulte [Ataques de sustitución de dependencias](#). Para protegerse de un ataque de sustitución de dependencias, configura los controles de origen del paquete en un repositorio para limitar la forma en que se pueden añadir las versiones de ese paquete al repositorio.

Deberías considerar la posibilidad de configurar los controles de origen de los paquetes para que las nuevas versiones de distintos paquetes procedan tanto de fuentes internas, como la publicación directa, como de fuentes externas, como los repositorios públicos. De forma predeterminada, los controles de origen de los paquetes se configuran en función de cómo se agrega la primera versión de un paquete al repositorio.

Configuración de control de origen del paquete

Con los controles de origen de los paquetes, puede configurar cómo se pueden añadir las versiones de los paquetes a un repositorio. Las siguientes listas incluyen la configuración y los valores de control de origen de los paquetes disponibles.

Publicación

Esta configuración configura si las versiones de los paquetes se pueden publicar directamente en el repositorio mediante administradores de paquetes o herramientas similares.

- **ALLOW:** las versiones de los paquetes se pueden publicar directamente.
- **BLOCK:** las versiones de los paquetes no se pueden publicar directamente.

Repositorios ascendentes

Esta configuración configura si las versiones de los paquetes pueden ingerirse desde repositorios públicos externos o conservarse desde repositorios originales cuando lo solicite un administrador de paquetes.

- **PERMITIR:** cualquier versión del paquete se puede conservar desde otros CodeCatalyst repositorios configurados como repositorios ascendentes o se puede ingerir desde una fuente pública con una conexión externa.
- **BLOCK:** Las versiones de los paquetes no se pueden conservar de otros CodeCatalyst repositorios configurados como repositorios ascendentes ni se pueden ingerir desde una fuente pública con una conexión externa.

Configuración de control de origen de paquetes predeterminada

Los controles de origen de los paquetes predeterminados se basarán en la forma en que se añada la primera versión de ese paquete al repositorio de paquetes.

- Si un administrador de paquetes publica directamente la primera versión del paquete, la configuración será Publish: **ALLOW** y Upstream: **BLOCK**.
- Si la primera versión del paquete proviene de una fuente pública, los ajustes serán Publish: **ALLOW** y Upstream: **BLOCK**.

Escenarios comunes de control de acceso a paquetes

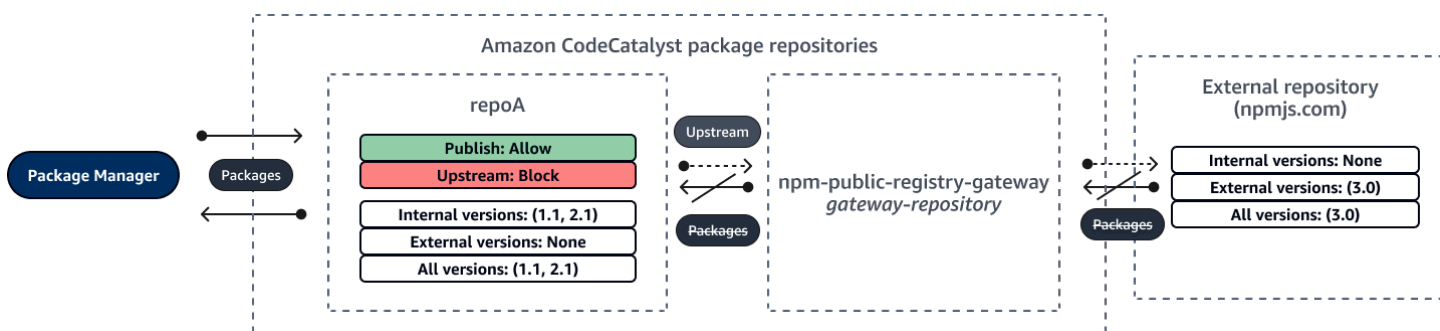
En esta sección se describen algunos escenarios comunes en los que se agrega una versión de CodeCatalyst paquete a un repositorio de paquetes. La configuración de control de origen del paquete se establece para los paquetes nuevos en función de cómo se añada la primera versión del paquete.

En los siguientes escenarios, un paquete interno se publica directamente desde un administrador de paquetes en su repositorio, como un paquete que usted mantiene. Un paquete externo es un paquete que existe en un repositorio público y que puede incorporarse a su repositorio mediante una conexión externa.

Se publica una versión de paquete externo para un paquete interno existente

En este escenario, consideremos un paquete interno, packageA. Tu equipo publica la primera versión del paquete del paquete A en un repositorio de CodeCatalyst paquetes. Como esta es la primera versión del paquete, la configuración del control de origen del paquete se establece automáticamente en Publish: Allow and Upstream: Block. Una vez publicado el paquete en tu repositorio, se publica un paquete con el mismo nombre en un repositorio público que está conectado a tu repositorio de CodeCatalyst paquetes. Podría tratarse de un intento de ataque de sustitución de dependencias contra el paquete interno o podría ser una coincidencia. En cualquier caso, los controles de origen de los paquetes están configurados para bloquear la ingesta de la nueva versión externa y protegerse así de un posible ataque.

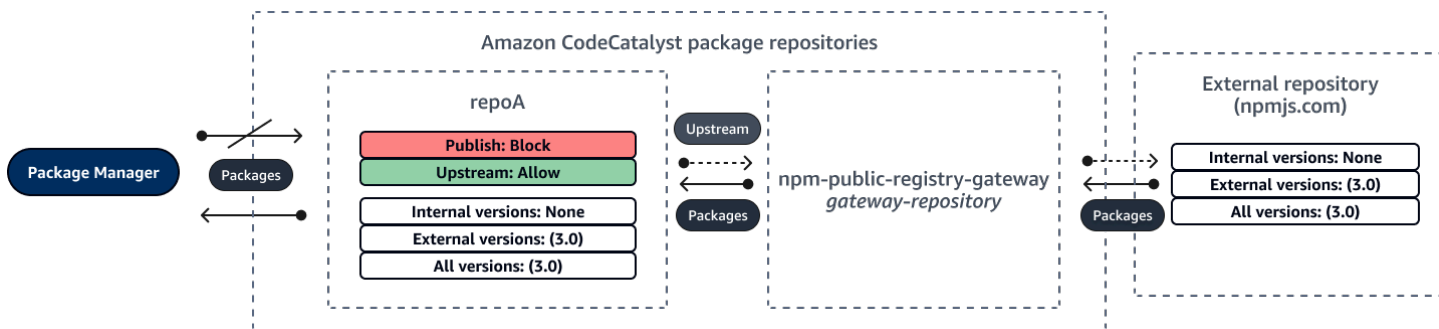
En la siguiente imagen, RePoA es su repositorio de CodeCatalyst paquetes con una conexión externa a un repositorio público. Su repositorio contiene las versiones 1.1 y 2.1 de packageA, pero la versión 3.0 está publicada en el repositorio público. Normalmente, RePoA ingiere la versión 3.0 después de que un administrador de paquetes solicite el paquete. Como la ingesta de paquetes está configurada como Bloquear, la versión 3.0 no se ingiere en su repositorio de CodeCatalyst paquetes y no está disponible para los administradores de paquetes conectados a él.



Se publica una versión de paquete interno para un paquete externo existente

En este escenario, un paquete, el paquete B, existe externamente en un repositorio público que usted ha conectado a su repositorio. Cuando un administrador de paquetes conectado a su repositorio solicita el packageB, la versión del paquete se introduce en su repositorio desde el repositorio público. Como esta es la primera versión de paquete de packageB que se agrega a su repositorio, los ajustes de origen del paquete están configurados como Publish: BLOCK y Upstream: ALLOW. Más adelante, intenta publicar una versión con el mismo nombre de paquete en el repositorio. Es posible que no conozcas el paquete público y estés intentando publicar un paquete no relacionado con el mismo nombre, que estés intentando publicar una versión parcheada o que estés intentando publicar directamente la versión exacta del paquete que ya existe externamente. CodeCatalyst rechaza la versión que está intentando publicar, pero puede anular el rechazo de forma explícita y publicar la versión, si es necesario.

En la imagen siguiente, RePoA es su repositorio de CodeCatalyst paquetes con una conexión externa a un repositorio público. El repositorio de paquetes contiene la versión 3.0 que ingirió del repositorio público. Desea publicar la versión 1.2 en su repositorio de paquetes. Normalmente, puede publicar la versión 1.2 en RePoA, pero como la publicación está configurada como Bloquear, la versión 1.2 no se puede publicar.



Publicar una versión parcheada de un paquete externo existente

En este escenario, un paquete, el paquete B, existe externamente en un repositorio público que usted ha conectado a su repositorio de paquetes. Cuando un administrador de paquetes conectado a su repositorio solicita el packageB, la versión del paquete se introduce en su repositorio desde el repositorio público. Como esta es la primera versión de paquete de packageB que se agrega a su repositorio, los ajustes de origen del paquete están configurados como Publish: BLOCK y Upstream: ALLOW. Tu equipo decide publicar las versiones parcheadas de este paquete en el repositorio. Para poder publicar las versiones de los paquetes directamente, su equipo cambia la configuración del control de origen de los paquetes a Publish: ALLOW y Upstream: BLOCK. Las versiones de este paquete ahora se pueden publicar directamente en su repositorio e ingerir desde los repositorios públicos. Una vez que su equipo publique las versiones de los paquetes parcheados, cambiará la configuración de origen del paquete a Publish: BLOCK and Upstream: ALLOW.

Edición de los controles de origen del paquete

Los controles de origen de los paquetes se configuran automáticamente en función de cómo se agrega la primera versión de un paquete al repositorio de paquetes. Para obtener más información, consulte [Configuración de control de origen de paquetes predeterminada](#). Para añadir o editar los controles de origen de un paquete en un repositorio de CodeCatalyst paquetes, lleve a cabo los pasos del siguiente procedimiento.

Para añadir o editar los controles de origen de los paquetes

1. En el panel de navegación, elija Packages (Paquetes).
2. Elija el repositorio de paquetes que contiene el paquete que desea editar.
3. En la tabla Paquetes, busque y elija el paquete que desee editar.
4. En la página de resumen del paquete, selecciona Editar controles de origen.
5. En los controles de origen, selecciona los controles de origen del paquete que deseas configurar para este paquete. Los dos ajustes de control de origen del paquete, Publish y Upstream, deben configurarse al mismo tiempo.
 - Para permitir la publicación directa de las versiones de los paquetes, en Publicar, seleccione Permitir. Para bloquear la publicación de versiones de paquetes, seleccione Bloquear.
 - Para permitir la ingesta de paquetes de repositorios externos y la extracción de paquetes de repositorios anteriores, en Recursos ascendentes, seleccione Permitir. Para bloquear la ingestión y extracción de versiones de paquetes desde repositorios externos y ascendentes, seleccione Bloquear.
6. Seleccione Guardar.

Repositorios editoriales y originales

En CodeCatalyst, no puede publicar versiones de paquetes que estén presentes en repositorios ascendentes o repositorios públicos accesibles. Por ejemplo, supongamos que desea publicar un paquete npm en un repositorio y `lodash@1.0` que `myrepo` tiene un repositorio ascendente con una conexión externa a `npmjs.com`. `myrepo` Considere los siguientes escenarios.

1. Los ajustes de control de origen de los paquetes de `lodash` son Publish: ALLOW y Upstream: ALLOW. Si `lodash@1.0` está presente en el repositorio principal o en `npmjs.com`, CodeCatalyst rechaza cualquier intento de publicar en él emitiendo un error 409 de conflicto. `myrepo` Aún puede publicar una versión diferente, por ejemplo `lodash@1.1`.

2. Los ajustes de control de origen de los paquetes de `lodash` son `Publish: ALLOW` y `Upstream: BLOCK`. Puede publicar en su repositorio cualquier versión de `lodash` que aún no exista porque no se puede acceder a las versiones de los paquetes.
3. Los ajustes de control de origen de los paquetes de `lodash` son `Publish: BLOCK` y `Upstream: ALLOW`. No puede publicar ninguna versión del paquete directamente en su repositorio.

Ataques de sustitución de dependencias

Los gestores de paquetes simplifican el proceso de empaquetar y compartir código reutilizable. Estos paquetes pueden ser paquetes privados desarrollados por una organización para usarlos en sus aplicaciones, o pueden ser públicos, generalmente paquetes de código abierto que se desarrollan fuera de una organización y se distribuyen en repositorios de paquetes públicos. Al solicitar paquetes, los desarrolladores confían en su administrador de paquetes para obtener nuevas versiones de sus dependencias. Los ataques de sustitución de dependencias, también conocidos como ataques de confusión de dependencias, aprovechan el hecho de que un administrador de paquetes normalmente no tiene forma de distinguir las versiones legítimas de un paquete de las versiones maliciosas.

Los ataques de sustitución de dependencias pertenecen a un subconjunto de ataques conocidos como ataques a la cadena de suministro de software. Un ataque a la cadena de suministro de software es un ataque que aprovecha las vulnerabilidades en cualquier punto de la cadena de suministro de software.

Un ataque de sustitución de dependencias puede dirigirse a cualquier persona que utilice tanto paquetes desarrollados internamente como paquetes extraídos de repositorios públicos. Los atacantes identifican los nombres de los paquetes internos y, a continuación, colocan estratégicamente el código malicioso con el mismo nombre en los repositorios de paquetes públicos. Normalmente, el código malicioso se publica en un paquete con un número de versión alto. Los administradores de paquetes obtienen el código malicioso de estas fuentes públicas porque creen que los paquetes maliciosos son las últimas versiones del paquete. Esto provoca una «confusión» o una «sustitución» entre el paquete deseado y el paquete malicioso, lo que hace que el código quede comprometido.

Para evitar los ataques de sustitución de dependencias, Amazon CodeCatalyst proporciona controles de origen de los paquetes. Los controles de origen de los paquetes son ajustes que controlan cómo se pueden añadir los paquetes a tus repositorios. Los controles se configuran automáticamente cuando se añade la primera versión de un paquete nuevo a un CodeCatalyst repositorio. Los

controles pueden garantizar que las versiones de los paquetes no puedan publicarse directamente en su repositorio ni ingerirse desde fuentes públicas, lo que le protege de los ataques de sustitución de dependencias. Para obtener más información sobre los controles de origen de los paquetes y cómo cambiarlos, consulte [Edición de los controles de origen del paquete](#).

Con npm

En estos temas se describe cómo puede utilizar npm el administrador de paquetes Node.js con CodeCatalyst.

Note

CodeCatalyst admite node v4.9.1 y versiones posteriores npm v5.0.0 y posteriores.

Temas

- [Configuración y uso de npm](#)
- [control de etiquetas npm](#)

Configuración y uso de npm

Para usarlo npm con CodeCatalyst, debe conectarse npm a su repositorio de paquetes y proporcionar un token de acceso personal (PAT) para la autenticación. Puedes ver las instrucciones para conectarte npm al repositorio de paquetes en la CodeCatalyst consola.

Contenido

- [Configurar npm con CodeCatalyst](#)
- [Instalación de paquetes npm desde un repositorio de paquetes CodeCatalyst](#)
- [Instalación de paquetes npm desde npmjs mediante CodeCatalyst](#)
- [Publicar paquetes npm en su repositorio de paquetes CodeCatalyst](#)
- [soporte de comandos npm](#)
 - [Comandos compatibles que interactúan con un repositorio de paquetes](#)
 - [Comandos del lado del cliente compatibles](#)
 - [Comandos admitidos](#)

Configurar npm con CodeCatalyst

Las siguientes instrucciones explican cómo autenticarse y conectarse npm a su repositorio de CodeCatalyst paquetes. Para obtener más información sobre npm, consulta la documentación [oficial de npm](#).

Para conectarse **npm** a su repositorio de paquetes CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Vaya a su proyecto.
3. En el panel de navegación, elija Packages (Paquetes).
4. Elija su repositorio de paquetes de la lista.
5. Seleccione Conectar al repositorio.
6. En Detalles de configuración, en Package manager client, elija npm client.
7. Elija su sistema operativo para ver los pasos de configuración correspondientes.
8. Se requiere un token de acceso personal (PAT) para autenticar npm. CodeCatalyst Si ya tiene un token, puede usarlo. Si no es así, puedes crear uno siguiendo estos pasos.
 - a. (Opcional): actualice el nombre y la fecha de caducidad de la PAT.
 - b. Elija Crear token.
 - c. Copie y almacene su PAT en un lugar seguro.

Warning

No podrá volver a ver ni copiar su PAT después de cerrar el cuadro de diálogo. Las credenciales deben ser de corta duración para minimizar el tiempo que un atacante puede utilizarlas tras apropiarse indebidamente de ellas.

9. Ejecute los siguientes comandos desde el directorio raíz de su proyecto para configurar npm con su repositorio de paquetes. Los comandos harán lo siguiente:
 - Cree un `.npmrc` archivo a nivel de proyecto si su proyecto no tiene uno.
 - Añada la información de punto final del repositorio de paquetes a su archivo a nivel de proyecto `.npmrc`.
 - Añada sus credenciales (PAT) a su archivo a nivel de usuario `.npmrc`.

Sustituya los valores siguientes.

 Note

Si está copiando las instrucciones de la consola, los valores de los siguientes comandos se actualizan automáticamente y no es necesario cambiarlos.

- Sustituya el *nombre* de CodeCatalyst usuario por su nombre de usuario.
- Sustituya *PAT* por su CodeCatalyst PAT.
- Sustituya *space_name por el nombre* de su CodeCatalyst espacio.
- Sustituya *proj_name por el nombre* de su proyecto. CodeCatalyst
- Sustituya *repo_name por el nombre del repositorio* de su paquete. CodeCatalyst

```
npm set registry=https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/ --location project
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:_authToken=username:PAT
```

Para npm 6 o versiones anteriores: para que npm siempre pase el token de autenticación CodeCatalyst, incluso en el caso de GET solicitudes, defina la variable de configuración always-auth de la siguiente manera. `npm config set`

```
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:always-auth=true --location project
```

Instalación de paquetes npm desde un repositorio de paquetes CodeCatalyst

Después de conectar npm a tu repositorio siguiendo los pasos que se indican [Configurar npm con CodeCatalyst](#), puedes ejecutar npm comandos en tu repositorio.

Puedes instalar un paquete npm que esté en tu repositorio de CodeCatalyst paquetes o en uno de sus repositorios anteriores con el comando. `npm install`

```
npm install Lodash
```

Instalación de paquetes npm desde npmjs mediante CodeCatalyst

Puede instalar paquetes npm desde npmjs.com a través de un repositorio configurándolo con una conexión ascendente al CodeCatalyst repositorio de puerta de enlace conectado a npmjs.com,, npm-public-registry-gateway Los paquetes instalados desde npmjs se ingieren y almacenan en el repositorio de la puerta de enlace y en el repositorio de paquetes descendente más alejado.

Para instalar paquetes desde npmjs

1. Si aún no lo ha hecho, configúrelo npm con su repositorio de CodeCatalyst paquetes siguiendo los pasos que se indican en. [Configurar npm con CodeCatalyst](#)
2. Comprueba que tu repositorio haya agregado el repositorio de puerta de enlace npm-public-registry-gateway,, como una conexión ascendente. Puede comprobar qué fuentes ascendentes se añaden o añadir npm-public-registry-gatewaycomo fuente ascendente siguiendo las instrucciones que se indican en el repositorio [Añadir un repositorio ascendente](#) y seleccionándolo. npm-public-registry-gateway
3. Instale los paquetes con el `npm install` comando.

```
npm install package_name
```

Para obtener más información sobre la solicitud de paquetes desde los repositorios originales, consulte. [Solicitar una versión de paquete con repositorios ascendentes](#)

Publicar paquetes npm en su repositorio de paquetes CodeCatalyst

Una vez que haya terminado [Configurar npm con CodeCatalyst](#), puede ejecutar npm los comandos.

Puede publicar un paquete npm en un repositorio de CodeCatalyst paquetes con el `npm publish` comando.

```
npm publish
```

Para obtener información sobre cómo crear paquetes npm, consulte [Creación de módulos de Node.js](#) en npm Docs.

soporte de comandos npm

En las siguientes secciones se resumen los npm comandos que admiten los repositorios de CodeCatalyst paquetes, además de enumerar los comandos específicos que no son compatibles.

Temas

- [Comandos compatibles que interactúan con un repositorio de paquetes](#)
- [Comandos del lado del cliente compatibles](#)
- [Comandos admitidos](#)

Comandos compatibles que interactúan con un repositorio de paquetes

En esta sección se enumeran los npm comandos en los que el npm cliente realiza una o más solicitudes al registro para el que está configurado (por ejemplo, `npm config set registry`). Se ha comprobado que estos comandos funcionan correctamente cuando se invocan en un repositorio de CodeCatalyst paquetes.

Comando	Descripción
bugs	Investiga la ubicación de la URL del rastreador de errores de un paquete y, a continuación, intenta abrirla.
ci	Instala un proyecto desde cero.
deprecate	Deja en desuso una versión de un paquete.
dist-tag	Modifica las etiquetas de distribución de paquetes.
docs	Averigua la ubicación de la URL de la documentación de un paquete y, a continuación, intenta abrirla mediante el parámetro <code>--browser config</code> .
doctor	Ejecuta un conjunto de comprobaciones para validar que su instalación de npm pueda administrar sus JavaScript paquetes.

Comando	Descripción
install	Instala un paquete.
install-ci-test	Instala un proyecto desde cero y ejecuta pruebas. Alias: <code>npm ci</code> . Este comando ejecuta <code>unnpm ci</code> , seguido inmediatamente de <code>unnpm test</code> .
install-test	Instala el paquete y ejecuta las pruebas. Ejecuta <code>unnpm install</code> , seguido inmediatamente de <code>unnpm test</code> .
outdated	Comprueba el registro configurado para determinar si algún paquete instalado está desactualizado.
ping	Hace ping al registro npm configurado o dado y verifica la autenticación.
publish	Publica una versión del paquete en el registro.
update	Averigua la ubicación de la URL del repositorio de un paquete y, a continuación, intenta abrirla mediante el parámetro <code>--browser config</code> .
view	Muestra metadatos del paquete. También se puede usar para imprimir las propiedades de los metadatos.

Comandos del lado del cliente compatibles

Estos comandos no requieren ninguna interacción directa con un repositorio de paquetes, por lo que CodeCatalyst no requiere nada que los respalde.

Comando	Descripción
bin (heredado)	Muestra el <code>bin</code> directorio npm.

Comando	Descripción
build	Crea un paquete.
cache	Manipula la caché de paquetes.
completion	Permite completar tabulaciones en todos los comandos de npm.
config	Actualiza el contenido de los archivos <code>npmrc</code> globales y de usuario.
dedupe	Busca en el árbol de paquetes local e intenta simplificar la estructura moviendo las dependencias más arriba en el árbol, de forma que puedan compartirse más eficazmente entre varios paquetes dependientes.
edit	Edita un paquete instalado. Selecciona una dependencia en el directorio de trabajo actual y abre el directorio del paquete en el editor predeterminado.
explore	Busca un paquete instalado. Genera un subshell en el directorio del paquete instalado especificado. Si se especifica un comando, se ejecuta en la subshell, que se cierra inmediatamente.
help	Obtiene ayuda sobre npm.
help-search	Busca en la documentación de ayuda de npm.
init	Crea un archivo <code>package.json</code> .
link	Enlaza simbólicamente un directorio de paquetes.
ls	Muestra los paquetes instalados.

Comando	Descripción
pack	Crea un tarball a partir de un paquete.
prefix	Muestra un prefijo. Este es el directorio principal más cercano que contiene un <code>package.json</code> archivo, a menos que también <code>-g</code> se especifique.
prune	Elimina los paquetes que no figuran en la lista de dependencias del paquete principal.
rebuild	Ejecuta el comando <code>npm build</code> en las carpetas coincidentes.
restart	Ejecuta los scripts de parada, reinicio e inicio de un paquete y los scripts previos y posterior es asociados.
root	Imprime el <code>node_modules</code> directorio efectivo para que quede estándar.
run-script	Ejecuta scripts de paquetes arbitrarios.
shrinkwrap	Bloquea las versiones dependientes para su publicación.
uninstall	Desinsta un paquete.

Comandos admitidos

Estos npm comandos no son compatibles con los repositorios de CodeCatalyst paquetes.

Comando	Descripción	Notas
access	Establece el nivel de acceso de los paquetes publicados.	CodeCatalyst usa un modelo de permisos diferente al del repositorio público de npmjs.

Comando	Descripción	Notas
adduser	Añade una cuenta de usuario de registro	CodeCatalyst utiliza un modelo de usuario diferente del repositorio público de npmjs.
audit	Realiza una auditoría de seguridad.	CodeCatalyst actualmente no vende datos sobre vulnerabilidades de seguridad.
hook	Administra los enlaces npm, lo que incluye agregar, eliminar, enumerar y actualizar.	CodeCatalyst actualmente no admite ningún mecanismo de notificación de cambios.
login	Autentica a un usuario. Este es un alias para npm adduser.	CodeCatalyst utiliza un modelo de autenticación diferente del repositorio público de npmjs. Para obtener más información, consulte Configurar npm con CodeCatalyst .
logout	Cierra la sesión del registro.	CodeCatalyst utiliza un modelo de autenticación diferente del repositorio público de npmjs. No hay forma de cerrar sesión en un CodeCatalyst repositorio, pero los tokens de autenticación caducan una vez transcurrido el tiempo de caducidad configurable. La duración predeterminada del token es de 12 horas.

Comando	Descripción	Notas
owner	Administra a los propietarios de los paquetes.	CodeCatalyst utiliza un modelo de permisos diferente al del repositorio público de npmjs.
profile	Cambia la configuración de su perfil de registro.	CodeCatalyst utiliza un modelo de usuario diferente del repositorio público de npmjs.
search	Busca en el registro paquetes que coincidan con los términos de búsqueda.	CodeCatalyst no admite el search comando.
star	Marca sus paquetes favoritos.	CodeCatalyst actualmente no admite ningún mecanismo de favoritos.
stars	Visualiza los paquetes marcados como favoritos.	CodeCatalyst actualmente no admite ningún mecanismo de favoritos.
team	Administra los equipos y las membresías de los equipos.	CodeCatalyst utiliza un modelo de membresía de usuarios y grupos que es diferente del repositorio público de npmjs.
token	Administra sus tokens de autenticación.	CodeCatalyst utiliza un modelo diferente para obtener los tokens de autenticación. Para obtener más información, consulte Configurar npm con CodeCatalyst .

Comando	Descripción	Notas
unpublish	Elimina un paquete del registro.	CodeCatalyst no admite la eliminación de una versión de paquete de un repositorio mediante el cliente npm. Puede eliminar un paquete en la consola.
whoami	Muestra el nombre de usuario de npm.	CodeCatalyst utiliza un modelo de usuario diferente del repositorio público de npmjs.

control de etiquetas npm

Los registros npm admiten etiquetas, que son alias de cadena para las versiones de los paquetes. Puede usar etiquetas para proporcionar un alias en lugar de usar números de versión. Por ejemplo, tienes un proyecto con varios flujos de desarrollo y utilizas una etiqueta diferente para cada flujo (por ejemplo, `stable`, `beta`, `dev`, `canary`). Para obtener más información, consulta [dist-tag](#) en npm Docs.

De forma predeterminada, npm usa la etiqueta `latest` para identificar la versión actual de un paquete. `npm install pkg` (sin especificador `@version @tag`) instala la última etiqueta. Por lo general, los proyectos solo usan la etiqueta más reciente para las versiones estables. Se utilizan otras etiquetas para las versiones inestables o preliminares.

Edición de etiquetas con el cliente npm

Los tres `npm dist-tag` comandos (`addrm`, `yls`) funcionan de la misma manera en los repositorios de CodeCatalyst paquetes que en el registro [npm predeterminado](#).

Etiquetas npm y repositorios ascendentes

Cuando npm solicita las etiquetas de un paquete y las versiones de ese paquete también están presentes en un repositorio principal, CodeCatalyst fusiona las etiquetas antes de devolverlas al cliente. Por ejemplo, un repositorio denominado R tiene un repositorio anterior denominado U. En la siguiente tabla se muestran las etiquetas de un paquete denominado `web-helper` que está presente en ambos repositorios.

Repositorio	Package name	Etiquetas de paquetes
R	web-helper	latest (alias de la versión 1.0.0)
U	web-helper	alpha (alias de la versión 1.0.1)

En este caso, cuando el cliente npm recupera las etiquetas del *web-helper* paquete del repositorio R, recibe tanto las etiquetas más recientes como las etiquetas alfa. Las versiones a las que apuntan las etiquetas no cambiarán.

Cuando la misma etiqueta está presente en el mismo paquete, tanto en el repositorio original como en el local, CodeCatalyst utiliza la etiqueta que se actualizó por última vez. Por ejemplo, supongamos que las etiquetas de webhelper se han modificado para que tengan el siguiente aspecto.

Repositorio	Package name	Etiquetas de paquetes	Última actualización
R	web-helper	latest (alias de la versión 1.0.0)	1 de enero de 2023
U	web-helper	latest (alias de la versión 1.0.1)	1 de junio de 2023

En este caso, cuando el cliente npm busque las etiquetas del paquete web-helper del repositorio R, la última etiqueta tendrá el alias de la versión 1.0.1 porque se actualizó por última vez. Esto facilita el consumo de nuevas versiones de paquetes en un repositorio principal que aún no están presentes en un repositorio local mediante la ejecución. `npm update`

Cuotas para paquetes

En la siguiente tabla se describen las cuotas y los límites de los paquetes en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Recurso	Cuota predeterminada
Repositorios de paquetes	Máximo de 1000 por espacio.
Repositorios ascendentes directos	Máximo de 10 por repositorio de paquetes.
Se buscaron repositorios de paquetes ascendentes	Se buscaron un máximo de 25 repositorios ascendentes por versión de paquete solicitada.
Tamaño del archivo de activos de Package	Máximo de 5 GB por paquete de activos.
Package assets	Máximo de 150 por versión de paquete.

Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst

Tras escribir el código de tu aplicación en [CodeCatalystun entorno](#) de desarrollo y subirlo a tu [repositorio de CodeCatalyst código fuente](#), estarás listo para implementarlo. La forma de hacerlo automáticamente es mediante un flujo de trabajo.

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Acerca del archivo de definición del flujo de trabajo

Un archivo de definición de flujo de trabajo es un archivo YAML que describe el flujo de trabajo. El archivo se almacena en una `~/ .codecatalyst/workflows/` carpeta en la raíz del [repositorio de origen](#). El archivo puede tener la extensión `.yaml` o `.yml`.

El siguiente es un ejemplo de un archivo de definición de flujo de trabajo sencillo. Explicamos cada línea de este ejemplo en la siguiente tabla.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

Actions:**Build:**

Identifier: aws/build@v1

Inputs:**Sources:**

- WorkflowSource

Configuration:**Steps:**

- Run: docker build -t MyApp:latest .

Línea	Descripción
<pre>Name: MyWorkflow</pre>	<p>Especifica el nombre del flujo de trabajo. Para obtener más información sobre la Name propiedad, consulte Propiedades de primer nivel.</p>
<pre>SchemaVersion: 1.0</pre>	<p>Especifica la versión del esquema de flujo de trabajo. Para obtener más información sobre la SchemaVersion propiedad, consulte Propiedades de primer nivel.</p>
<pre>RunMode: QUEUED</pre>	<p>Indica cómo CodeCatalyst gestiona las ejecuciones múltiples. Para obtener más información sobre el modo de ejecución, consulte Configuración de ejecuciones en cola, reemplazadas y paralelas.</p>
<pre>Triggers:</pre>	<p>Especifica la lógica que hará que se inicie la ejecución de un flujo de trabajo. Para obtener más información acerca de los disparadores, consulte Trabajar con factores desencadenantes.</p>
<pre>- Type: PUSH Branches: - main</pre>	<p>Indica que el flujo de trabajo debe iniciarse siempre que insertes código main en la rama del repositorio fuente predeterminado. Para obtener más información sobre la fuente del flujo de trabajo, consulte Trabajar con fuentes.</p>

Línea	Descripción
<pre>Actions:</pre>	<p>Define las tareas que se van a realizar durante la ejecución de un flujo de trabajo. En este ejemplo, la Actions sección define una sola acción llamada <code>Build</code>. Para obtener más información sobre las acciones, consulte Trabajar con acciones.</p>
<pre>Build:</pre>	<p>Define las propiedades de la <code>Build</code> acción. Para obtener más información sobre la acción de creación, consulte Creación mediante flujos de trabajo en CodeCatalyst.</p>
<pre>Identifier: aws/builddev1</pre>	<p>Especifica el identificador único y codificado de la acción de creación.</p>
<pre>Inputs: Sources: - WorkflowSource</pre>	<p>Indica que la acción de compilación debe buscar en el repositorio de <code>WorkflowSource</code> origen los archivos que necesita para completar su procesamiento. Para obtener más información, consulte Trabajar con fuentes.</p>
<pre>Configuration:</pre>	<p>Contiene las propiedades de configuración específicas de la acción de compilación.</p>
<pre>Steps: - Run: docker build -t MyApp:latest .</pre>	<p>Indica a la acción de compilación que debe crear una imagen de Docker llamada <code>MyApp</code> y etiquetarla. <code>latest</code></p>

Para obtener una lista completa de todas las propiedades disponibles en el archivo de definición del flujo de trabajo, consulte. [Referencia de definición de flujo de trabajo](#)

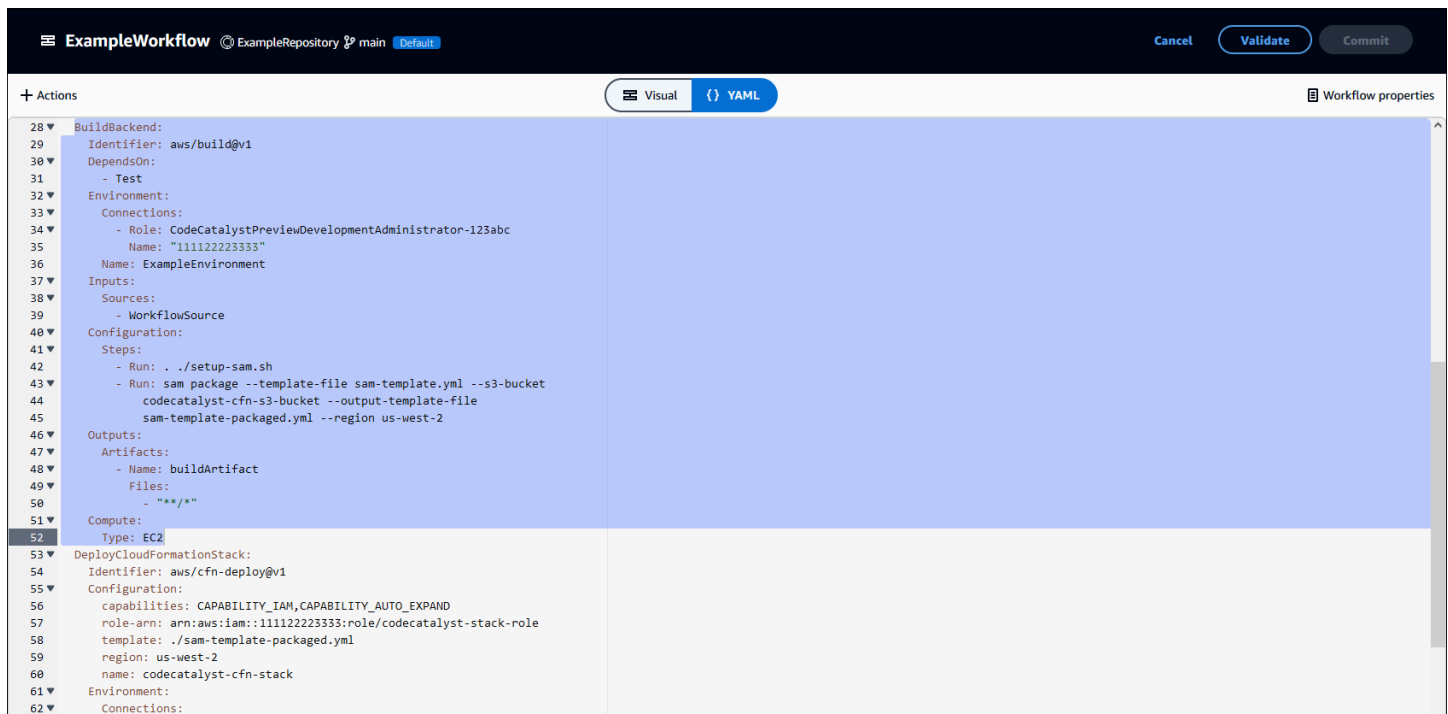
Uso de los CodeCatalyst editores visuales y YAML de la consola

Para crear y editar el archivo de definición del flujo de trabajo, puedes usar el editor que prefieras, pero te recomendamos usar el editor visual o el editor YAML de la CodeCatalyst consola. Estos

editores ofrecen una útil validación de archivos para garantizar que los nombres de las propiedades, los valores, el anidamiento, el espaciado, el uso de mayúsculas, etc. de YAML sean correctos.

La siguiente imagen muestra un flujo de trabajo en el editor visual. El editor visual le ofrece una interfaz de usuario completa a través de la cual puede crear y configurar el archivo de definición del flujo de trabajo. El editor visual incluye un diagrama de flujo de trabajo (1) que muestra los componentes principales del flujo de trabajo y un área de configuración (2).

Como alternativa, puedes usar el editor YAML, que se muestra en la siguiente imagen. Usa el editor YAML para pegar bloques de código grandes (de un tutorial, por ejemplo) o para añadir propiedades avanzadas que no se ofrecen en el editor visual.

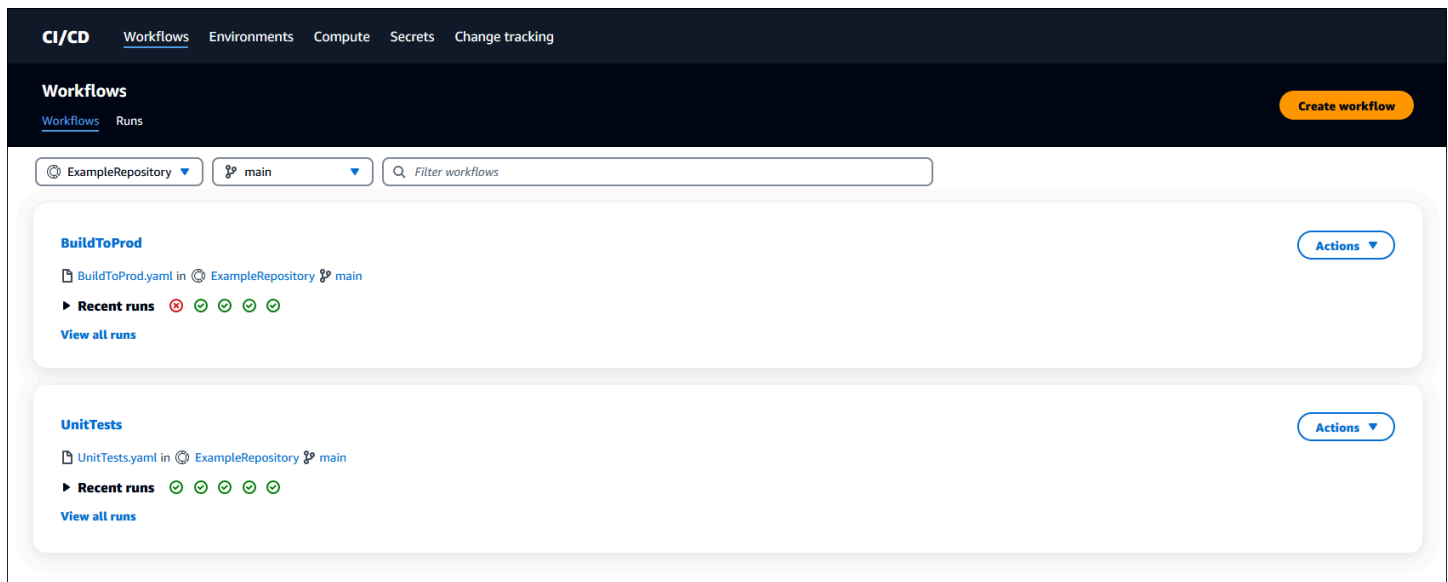


Puedes cambiar del editor visual al editor YAML para ver el efecto que tienen tus configuraciones en el código YAML subyacente.

Descubriendo flujos de trabajo

Puedes ver tu flujo de trabajo en la página de resumen de los flujos de trabajo, junto con otros flujos de trabajo que hayas configurado en el mismo proyecto.

La siguiente imagen muestra la página de resumen de los flujos de trabajo. Se rellena con dos flujos de trabajo: BuildToPrody UnitTests. Puede ver que ambos se han ejecutado varias veces. Puedes elegir Ejecuciones recientes para ver rápidamente el historial de ejecuciones o elegir el nombre del flujo de trabajo para ver el código YAML del flujo de trabajo y otra información detallada.



Ver los detalles de ejecución del flujo de trabajo

Para ver los detalles de una ejecución de flujo de trabajo, seleccione la ejecución en la página de resumen de los flujos de trabajo.

La siguiente imagen muestra los detalles de una ejecución de flujo de trabajo llamada Run-CC11d que se inició automáticamente al confirmarse con la fuente. El diagrama de flujo de trabajo indica que se ha producido un error en una acción (1). Puede navegar hasta los registros (2) para ver los mensajes de registro detallados y solucionar problemas. Para obtener más información sobre las ejecuciones del flujo de trabajo, consulte [Trabajar con corridas](#).

The screenshot displays the Amazon CodeCatalyst interface for a workflow named 'Run-cc11d'. The workflow is in a 'Failed' state, with a 'BuildBackend' action highlighted as the failed action. The 'BuildBackend' action is shown in a 'Failed' state with a duration of 1 minute 10 seconds. The detailed log message for the failed action is as follows:

```

1 [Container] 2023/04/05 15:04:37 Running command sam package --template-file sam-temp
2
3 SAM CLI now collects telemetry to better understand customer needs.
4
5 You can OPT OUT and disable telemetry collection by setting the
6 environment variable SAM_CLI_TELEMETRY=0 in your shell.
7 Thanks for your help!
8
9 Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/develop
10
11 Error: Template file not found at /codecatalyst/output/src3862/src/git-codecommit.us

```

Siguientes pasos

Para obtener más información sobre los conceptos de flujos de trabajo, consulte [Conceptos de flujos de](#)

Para crear su primer flujo de trabajo, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).

Conceptos de flujos de

Estos son algunos conceptos y términos que debes conocer a la hora de crear, probar o implementar tu código con flujos de trabajo integrados CodeCatalyst.

Flujos de trabajo

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Para obtener más información acerca de los flujos de trabajo, consulte [Trabajar con flujos de trabajo](#).

Archivos de definición del flujo de trabajo

Un archivo de definición de flujo de trabajo es un archivo YAML que describe tu flujo de trabajo. El archivo se almacena en una `~/codecatalyst/workflows/` carpeta en la raíz del [repositorio de origen](#). El archivo puede tener la extensión `.yaml` o `.yml`.

Para obtener más información sobre el archivo de definición del flujo de trabajo, consulte [Referencia de definición de flujo de trabajo](#)

Acciones

Una acción es el componente principal de un flujo de trabajo y define una unidad lógica de trabajo que se debe realizar durante la ejecución de un flujo de trabajo. Por lo general, un flujo de trabajo incluye varias acciones que se ejecutan de forma secuencial o paralela en función de cómo las haya configurado.

Para obtener más información sobre las acciones, consulte [Trabajar con acciones](#).

Grupos de acciones

Un grupo de acciones contiene una o más acciones. Agrupar las acciones en grupos de acciones le ayuda a mantener el flujo de trabajo organizado y también le permite configurar las dependencias entre los distintos grupos de acciones.

Para obtener más información sobre los grupos de acciones, consulte [Agrupación de acciones en grupos de acciones](#)

Artefactos

Un artefacto es el resultado de una acción de flujo de trabajo y, por lo general, consiste en una carpeta o un archivo de archivos. Los artefactos son importantes porque permiten compartir archivos e información entre acciones.

Para obtener más información acerca de los artefactos, consulte [Trabajar con artefactos](#).

Cálculo

La computación se refiere al motor informático (la CPU, la memoria y el sistema operativo) que administra y mantiene CodeCatalyst para ejecutar las acciones del flujo de trabajo.

Para obtener más información sobre el procesamiento, consulte [Trabajar con computación](#).

Entornos

Un entorno, que no debe confundirse con un [entorno de desarrollo](#), es el lugar en el que se despliega el código. Por lo general, contiene una instancia de una aplicación en ejecución junto con su infraestructura asociada. Puede asignar un nombre a su entorno, como desarrollo, prueba, puesta en escena o producción. Todas las implementaciones generadas por CodeCatalyst un entorno aparecerán en la página Entornos. Para configurar un entorno, debe asignarle un nombre, por ejemplo `plomy-production-environment`, y luego asociarlo al suyo Cuenta de AWS.

Además de mostrar la información de implementación, un entorno también sirve como mecanismo mediante el cual se asignan funciones de AWS IAM a [las acciones](#) del flujo de trabajo.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#).

Informes

Un informe contiene detalles sobre las pruebas que se realizan durante la ejecución de un flujo de trabajo. Puede crear informes, como un informe de prueba, un informe de cobertura de código, un informe de análisis de la composición del software y un informe de análisis estático. Puede usar un informe para ayudar a solucionar un problema durante un flujo de trabajo. Si tiene muchos informes de varios flujos de trabajo, puede utilizarlos para ver las tendencias y los índices de errores, lo que le ayudará a optimizar las aplicaciones y las configuraciones de implementación.

Para obtener más información acerca de los informes, consulte [Tipos de informes de prueba](#).

Ejecuciones

Una ejecución es una iteración única de un flujo de trabajo. Durante una ejecución, CodeCatalyst realiza las acciones definidas en el archivo de configuración del flujo de trabajo y genera los registros, artefactos y variables asociados.

Para obtener más información sobre las ejecuciones, consulte [Trabajar con corridas](#).

Orígenes

Una fuente, también denominada fuente de entrada, es un repositorio de fuentes al que necesita acceder una [acción de flujo](#) de trabajo para llevar a cabo sus tareas. Por ejemplo, es posible que una acción de flujo de trabajo necesite acceder a una fuente para obtener pruebas unitarias y ejecutarlas en los archivos fuente de la aplicación.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Variables

Una variable es un par clave-valor que contiene información a la que puede hacer referencia en su CodeCatalyst flujo de trabajo.

Para obtener más información sobre las variables, consulte [Trabajar con variables](#).

Activadores del flujo de

Un activador de flujo de trabajo, o simplemente un activador, permite iniciar la ejecución automática de un flujo de trabajo cuando se producen determinados eventos, como una inserción de código. Es posible que desee configurar los activadores para evitar que los desarrolladores de software tengan que iniciar manualmente las ejecuciones del flujo de trabajo a través de la CodeCatalyst consola.

Puede utilizar tres tipos de activadores:

- Empujar: un activador de inserción de código hace que se inicie la ejecución de un flujo de trabajo cada vez que se ejecuta una confirmación.
- Solicitud de extracción: cuando se activa una solicitud de extracción, se inicia la ejecución de un flujo de trabajo cada vez que se crea, revisa o cierra una solicitud de extracción.
- Programación: un activador de programación hace que la ejecución de un flujo de trabajo comience según el cronograma que tú definas. Considere la posibilidad de utilizar un activador

programado para ejecutar versiones nocturnas de su software, de modo que la última versión esté lista para que los desarrolladores de software puedan trabajar en ella a la mañana siguiente.

Puedes usar activadores push, pull request y programables solos o combinados en el mismo flujo de trabajo.

Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).

Introducción a los flujos de trabajo en CodeCatalyst

En este tutorial, aprenderás a crear y configurar tu primer flujo de trabajo.

Tip

¿Prefieres empezar con un flujo de trabajo preconfigurado? Consulte [Crear un proyecto con un plano](#), que incluye instrucciones para configurar un proyecto con un flujo de trabajo que funcione, una aplicación de muestra y otros recursos.

Temas

- [Requisitos previos](#)
- [Paso 1: Crea y configura tu flujo de trabajo](#)
- [Paso 2: Guarda tu flujo de trabajo con una confirmación](#)
- [Paso 3: Ver los resultados de la ejecución](#)
- [\(Opcional\) Paso 4: limpiar](#)

Requisitos previos

Antes de empezar

- Necesitas un CodeCatalyst espacio. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- En tu CodeCatalyst espacio, necesitas un CodeCatalyst proyecto vacío para empezar desde cero llamado:

```
codecatalyst-project
```

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

- En tu proyecto, necesitas un CodeCatalyst repositorio llamado:

```
codecatalyst-source-repository
```

Para obtener más información, consulte [Crear un repositorio de código fuente](#).

Note

Si tienes un proyecto y un repositorio fuente existentes, puedes usarlos; sin embargo, al crear otros nuevos, la limpieza será más fácil al final de este tutorial.

Paso 1: Crea y configura tu flujo de trabajo

En este paso, se crea y configura un flujo de trabajo que compila y prueba automáticamente el código fuente cuando se realizan cambios.

Para crear tu flujo de trabajo

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Seleccione Crear flujo de trabajo.

El archivo de definición del flujo de trabajo aparece en el editor YAML de la CodeCatalyst consola.

Para configurar el flujo de trabajo

Puedes configurar tu flujo de trabajo en el editor visual o en el editor YAML. Empecemos con el editor YAML y luego pasemos al editor visual.

1. Selecciona + Acciones para ver una lista de acciones de flujo de trabajo que puedes añadir a tu flujo de trabajo.

2. En la acción Crear, selecciona + para añadir el YAML de la acción al archivo de definición del flujo de trabajo. Tu flujo de trabajo ahora tiene un aspecto similar al siguiente.

```
Name: Workflow_fe47
SchemaVersion: "1.0"

# Optional - Set automatic triggers.
Triggers:
  - Type: Push
    Branches:
      - main

# Required - Define action configurations.
Actions:
  Build_f0:
    Identifier: aws/build@v1

    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this workflow as
a source

    Outputs:
      AutoDiscoverReports:
        Enabled: true
        # Use as prefix for the report files
        ReportNamePrefix: rpt

    Configuration:
      Steps:
        - Run: echo "Hello, World!"
        - Run: echo "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>" >> report.xml
        - Run: echo "<testsuite tests=\"1\" name=\"TestAgentJunit\" >" >>
report.xml
        - Run: echo "<testcase classname=\"TestAgentJunit\" name=\"Dummy
Test\"/></testsuite>" >> report.xml
```

El flujo de trabajo copia los archivos del repositorio de WorkflowSource origen en la máquina de procesamiento que ejecuta la Build_f0 acción, los imprime Hello, World! en los registros, descubre los informes de las pruebas en la máquina de procesamiento y los envía a la página de informes de la CodeCatalyst consola.

3. Selecciona Visual para ver el archivo de definición del flujo de trabajo en el editor visual. Los campos del editor visual permiten configurar las propiedades de YAML que se muestran en el editor de YAML.

Paso 2: Guarda tu flujo de trabajo con una confirmación

En este paso, guardas los cambios. Como los flujos de trabajo se almacenan como `.yaml` archivos en el repositorio, los cambios se guardan con confirmaciones.

Para confirmar los cambios de tu flujo de trabajo

1. (Opcional) Selecciona Validar para asegurarte de que el código YAML del flujo de trabajo sea válido.
2. Elija Commit (Confirmar).
3. En Nombre del archivo de flujo de trabajo, introduce un nombre para el archivo de configuración del flujo de trabajo, por ejemplo **my-first-workflow**.
4. En Mensaje de confirmación, introduce un mensaje para identificar tu confirmación, por ejemplo **create my-first-workflow.yaml**.
5. En Repositorio, elige el repositorio en el que quieres guardar el flujo de trabajo (`codecatalyst-repository`).
6. En Nombre de sucursal, elige la rama en la que quieres guardar el flujo de trabajo (`main`).
7. Elija Commit (Confirmar).

El nuevo flujo de trabajo aparece en la lista de flujos de trabajo. Es posible que tarde varios minutos en aparecer.

Como los flujos de trabajo se guardan con confirmaciones y porque el flujo de trabajo tiene configurado un activador de inserción de código, al guardar el flujo de trabajo se inicia un flujo de trabajo que se ejecuta automáticamente.

Paso 3: Ver los resultados de la ejecución

En este paso, accedes a la ejecución que se inició desde tu confirmación y ves los resultados.

Para ver los resultados de la ejecución

1. Elija el nombre de su flujo de trabajo, por ejemplo, `Workflow_fe47`.

Un diagrama de flujo de trabajo que muestre la etiqueta del repositorio de origen (WorkflowSource) y la acción de creación (por ejemplo, build_F0).

2. En el diagrama de ejecución del flujo de trabajo, elige la acción de creación (por ejemplo, build_F0).
3. Revise el contenido de las pestañas Registros, Informes, Configuración y Variables. Estas pestañas muestran los resultados de la acción de creación.

Para obtener más información, consulte [Visualización de los resultados de una acción de construcción](#).

(Opcional) Paso 4: limpiar

En este paso, limpiará los recursos que creaste en este tutorial.

Para eliminar recursos

1. Si ha creado un proyecto nuevo para este tutorial, elimínelo. Para ver instrucciones, consulte [Eliminar un proyecto en Amazon CodeCatalyst](#). Al eliminar el proyecto, también se eliminan el repositorio de origen y el flujo de trabajo.
2. Elimine el flujo de trabajo si aún no se ha eliminado. Para ver instrucciones, consulte [Para eliminar un flujo de trabajo](#).
3. Elimine el repositorio si aún no se ha eliminado. Para obtener instrucciones, consulte [Eliminar un repositorio de fuentes](#).

Visualización de la calidad del código y el estado de la implementación por confirmación CodeCatalyst

En cualquier momento del ciclo de vida del desarrollo, es importante conocer el estado de despliegue de determinadas confirmaciones, como las correcciones de errores, las nuevas funciones u otros cambios importantes. Tenga en cuenta los siguientes escenarios en los que la función de seguimiento del estado de la implementación resulta útil para los equipos de desarrollo:


- Como desarrollador, has realizado una corrección para corregir un error y quieres informar del estado de su despliegue en todos los entornos de despliegue de tu equipo.

- Como gestor de versiones, quieres ver una lista de las confirmaciones implementadas para hacer un seguimiento de su estado de despliegue e informar al respecto.

CodeCatalyst proporciona una vista que puede utilizar para determinar de un vistazo dónde se han implementado las confirmaciones o los cambios individuales y en qué entorno. Esta vista incluye:

- Una lista de confirmaciones.
- El estado de las implementaciones que incluyen las confirmaciones.
- Los entornos en los que las confirmaciones se implementan correctamente.
- El estado de cualquier prueba que se ejecute en función de las confirmaciones en su flujo de trabajo de CI/CD.

El siguiente procedimiento detalla cómo navegar hasta esta vista y utilizarla para realizar un seguimiento de los cambios en el proyecto.

 Note

El seguimiento del estado de la implementación por confirmación solo se admite en los [CodeCatalyst repositorios](#). No puedes usar esta función con un [GitHub repositorio](#).

Para realizar un seguimiento del estado de la implementación mediante confirmación


1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, selecciona CI/CD y, a continuación, selecciona Cambiar seguimiento.
4. En las dos listas desplegadas de la parte superior del panel principal, selecciona el repositorio y la rama de origen que contienen las confirmaciones cuyo estado de publicación deseas ver.
5. Selecciona Ver cambios.

Aparece una lista de confirmaciones.

Para cada confirmación, puedes ver lo siguiente:

- Información de confirmación, como el ID, el autor, el mensaje y cuándo se confirmó. Para obtener más información, consulte [Repositorios de fuentes en CodeCatalyst](#).

- El estado de las implementaciones en cada entorno. Para obtener más información, consulte [Trabajo con entornos](#).
- Resultados de la cobertura de las pruebas y el código. Para obtener más información, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#).

 Note

No se muestran los resultados del análisis de composición del software (SCA).

6. (Opcional) Para ver más información sobre los cambios relacionados con una confirmación específica, incluida la última implementación y la cobertura detallada del código y la información sobre las pruebas unitarias, seleccione Ver detalles de esa confirmación.

Creación mediante flujos de trabajo en CodeCatalyst

Con [CodeCatalyst los flujos de trabajo](#), puede crear aplicaciones y otros recursos.

Temas

- [¿Cómo puedo crear una aplicación?](#)
- [Ventajas de la acción de creación](#)
- [Alternativas a la acción de construcción](#)
- [Añadir la acción de construcción](#)
- [Visualización de los resultados de una acción de construcción](#)
- [Tutorial: Carga artefactos a Amazon S3](#)

¿Cómo puedo crear una aplicación?

Para crear una aplicación o un recurso CodeCatalyst, primero debe crear un flujo de trabajo y, a continuación, especificar una acción de creación en su interior.

Una acción de compilación es un componente básico del flujo de trabajo que compila el código fuente, ejecuta pruebas unitarias y produce artefactos listos para su despliegue.

Puedes añadir una acción de compilación a tu flujo de trabajo mediante el editor visual o el editor YAML de la CodeCatalyst consola.

Los pasos generales para crear una aplicación o un recurso son los siguientes.

Para crear una aplicación (tareas de alto nivel)

1. En CodeCatalyst, agregas el código fuente de la aplicación que deseas crear. Para obtener más información, consulte [Trabajar con repositorios de código fuente en CodeCatalyst](#).
2. En CodeCatalyst, creas un flujo de trabajo. En el flujo de trabajo se define cómo crear, probar e implementar la aplicación. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).
3. (Opcional) En el flujo de trabajo, se agrega un activador que indica los eventos que harán que el flujo de trabajo se inicie automáticamente. Para más información, consulte [Trabajar con factores desencadenantes](#)
4. En el flujo de trabajo, se agrega una acción de compilación que compila y empaqueta el código fuente de la aplicación o el recurso. Si lo desea, también puede hacer que la acción de compilación ejecute pruebas unitarias, genere informes e implemente la aplicación si no desea utilizar una acción de prueba o despliegue para estos fines. Para obtener más información sobre las acciones de prueba e implementación, consulte [Añadir la acción de construcción](#).
5. (Opcional) En el flujo de trabajo, añade una acción de prueba y una acción de despliegue para probar e implementar la aplicación o el recurso. Puede elegir entre varias acciones preconfiguradas para implementar la aplicación en diferentes destinos, como Amazon ECS. Para obtener más información, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#) y [Implementación mediante flujos de trabajo en CodeCatalyst](#).
6. El flujo de trabajo se inicia de forma manual o automática mediante un disparador. El flujo de trabajo ejecuta las acciones de creación, prueba e implementación en secuencia para crear, probar e implementar la aplicación y los recursos en el destino. Para obtener más información, consulte [Iniciar una ejecución de flujo de trabajo](#).

Ventajas de la acción de creación

El uso de la acción de creación en un flujo de trabajo tiene las siguientes ventajas:

- Totalmente gestionado: la acción de compilación elimina la necesidad de configurar, aplicar parches, actualizar y administrar sus propios servidores de compilación.
- Bajo demanda: la acción de creación se amplía según la demanda para satisfacer sus necesidades de creación. Solo se paga por el número de minutos de compilación consumidos. Para obtener más información, consulte [Trabajar con computación](#).

- Listas para usar: CodeCatalyst incluye imágenes de Docker del entorno de ejecución preempaquetadas que se utilizan para ejecutar todas las acciones del flujo de trabajo, incluidas las acciones de compilación. Estas imágenes vienen preconfiguradas con herramientas útiles para crear aplicaciones, como Node.js y AWS CLI. Puede configurarlo CodeCatalyst para usar una imagen de compilación que suministre desde un registro público o privado. Para obtener más información, consulte [Trabajar con imágenes de Docker en un entorno de ejecución](#).

Alternativas a la acción de construcción

Si utilizas una acción de compilación para implementar tu aplicación, considera usar una acción de CodeCatalyst implementación en su lugar. Las acciones de despliegue realizan behind-the-scenes configuraciones que, de otro modo, tendrías que escribir manualmente si utilizas una acción de compilación. Para obtener más información sobre las acciones de despliegue disponibles, consulte [Lista de acciones de despliegue](#).

También puede utilizarlos AWS CodeBuild para crear sus aplicaciones. Para obtener más información, consulte [¿Qué es CodeBuild?](#) .

Temas

- [Añadir la acción de construcción](#)
- [Visualización de los resultados de una acción de construcción](#)
- [Tutorial: Carga artefactos a Amazon S3](#)

Añadir la acción de construcción

Utilice el siguiente procedimiento para añadir una acción de creación a su CodeCatalyst flujo de trabajo.

Visual

Para añadir una acción de creación mediante el editor visual

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

4. Elija Editar.
5. Elija Visual.
6. Elija Acciones.
7. En Acciones, selecciona Construir.
8. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acciones de compilación y prueba](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
9. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir una acción de compilación mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
4. Elija Editar.
5. Elige YAML.
6. Elija Acciones.
7. En Acciones, selecciona Construir.
8. Modifica las propiedades del código YAML según tus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acciones de compilación y prueba](#)
9. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Cree una definición de acción

La acción de compilación se define como un conjunto de propiedades YAML dentro del archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte [Referencia de acciones de compilación y prueba](#) la [Referencia de definición de flujo de trabajo](#).

Visualización de los resultados de una acción de construcción

Siga las instrucciones siguientes para ver los resultados de una acción de compilación, incluidos los registros, los informes y las variables generados.

Para ver los resultados de una acción de compilación

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
3. En el diagrama de flujo de trabajo, elige el nombre de la acción de compilación, por ejemplo, Construir.
4. Para ver los registros de la ejecución de la compilación, selecciona Registros. Se muestran los registros de las distintas fases de compilación. Puede ampliar los registros según sea necesario.
5. Para ver los informes de prueba generados por la acción de creación, elija Informes o, en el panel de navegación, elija Informes. Para obtener más información, consulte [Tipos de informes de prueba](#).
6. Para ver la configuración utilizada para la acción de compilación, elija Configuración. Para obtener más información, consulte [Añadir la acción de construcción](#).
7. Para ver las variables utilizadas por la acción de compilación, elija Variables. Para obtener más información, consulte [Trabajar con variables](#).

Tutorial: Carga artefactos a Amazon S3

En este tutorial, aprenderá a cargar artefactos a un bucket de Amazon S3 mediante un CodeCatalyst [flujo de trabajo](#) que incluye un par de [acciones de compilación](#). Estas acciones se ejecutan en serie cuando se inicia el flujo de trabajo. La primera acción de construcción genera dos archivos `Hello.txt` y `Goodbye.txt` los agrupa en un artefacto de construcción. La segunda acción de construcción carga el artefacto en Amazon S3. Configuraré el flujo de trabajo para que se ejecute cada vez que envíe una confirmación a su repositorio de origen.

Temas

- [Requisitos previos](#)
- [Paso 1: Crea un AWS rol](#)
- [Paso 2: Crear un bucket de Amazon S3](#)
- [Paso 3: Crea un repositorio de código fuente](#)
- [Paso 4: Crear un flujo de trabajo](#)
- [Paso 5: Verificar los resultados](#)
- [Limpieza](#)

Requisitos previos

Antes de comenzar, necesitará lo siguiente:

- Necesitas un CodeCatalyst espacio con una AWS cuenta conectada. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- En tu espacio, necesitas un CodeCatalyst proyecto vacío para empezar desde cero llamado:

```
codecatalyst-artifact-project
```

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

- En tu proyecto, necesitas un CodeCatalyst entorno llamado:

```
codecatalyst-artifact-environment
```

Configure este entorno de la siguiente manera:

- Elija cualquier tipo, como Desarrollo.
- Conecta tu AWS cuenta a ella.

Para obtener más información, consulte [Trabajo con entornos](#).

Paso 1: Crea un AWS rol

En este paso, crea un rol de AWS IAM que luego asignará a la acción de creación de su flujo de trabajo. Este rol otorga a la acción de CodeCatalyst construcción el permiso para acceder a tu AWS

cuenta y escribir en Amazon S3, donde se almacenará tu artefacto. La función se denomina función de creación.

Note

Si ya tienes un rol de compilación que creaste para otro tutorial, también puedes usarlo para este tutorial. Solo asegúrate de que tiene los permisos y la política de confianza que se muestran en el siguiente procedimiento.

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

Para crear un rol de creación

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Elija Crear política.
 - e. Seleccione la pestaña JSON.
 - f. Elimina el código existente.
 - g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-s3-build-policy
```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

- 2. Cree el rol de compilación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



```
]
}
```

- e. Elija Siguiente.
- f. En Políticas de permisos, busque `codecatalyst-s3-build-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

```
codecatalyst-s3-build-role
```

- i. En la descripción del rol, introduzca:

```
CodeCatalyst build role
```

- j. Seleccione Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

Paso 2: Crear un bucket de Amazon S3

En este paso, crearás un bucket de Amazon S3 en el `Hello.txt` que se cargarán los `Goodbye.txt` artefactos.

Para crear un bucket de Amazon S3

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. En el panel principal, selecciona Crear depósito.
3. En el nombre del bucket, introduce:

```
codecatalyst-artifact-bucket
```

4. En AWS Región, seleccione una región. En este tutorial se asume que ha elegido US West (Oregon) `us-west-2`. Para obtener información sobre las regiones compatibles con Amazon S3, consulte los [puntos de enlace y las cuotas de Amazon Simple Storage Service](#) en. Referencia general de AWS
5. En la parte inferior de la página, selecciona Crear depósito.
6. Copia el nombre del depósito que acabas de crear, por ejemplo:

```
codecatalyst-artifact-bucket
```

Ahora ha creado un bucket denominado **codecatalyst-artifact-bucket** en la región us-west-2 de EE. UU. Oeste (Oregón).

Paso 3: Crea un repositorio de código fuente

En este paso, crearás un repositorio de fuentes en CodeCatalyst. Este repositorio se utiliza para almacenar el archivo de definición del flujo de trabajo del tutorial.

Para obtener más información sobre los repositorios de fuentes, consulte [Crear un repositorio de código fuente](#).

Para crear un repositorio de origen

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto, `codecatalyst-artifact-project`.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
5. En Nombre del repositorio, ingresa:

```
codecatalyst-artifact-source-repository
```

6. Seleccione Crear.

Ahora ha creado un repositorio llamado `codecatalyst-artifact-source-repository`.

Paso 4: Crear un flujo de trabajo

En este paso, se crea un flujo de trabajo que consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información sobre los activadores, consulte [Trabajar con factores desencadenantes](#).

- Una acción de compilación denominada `GenerateFiles`: Al activarse, la `GenerateFiles` acción crea dos archivos `Hello.txt` y `Goodbye.txt` los empaqueta en un artefacto de salida denominado `codecatalystArtifact`.
- Otra acción de compilación denominada `Upload`: Al finalizar la `GenerateFiles` acción, la `Upload` acción ejecuta el AWS CLI comando `aws s3 sync` para cargar los archivos del repositorio de origen `codecatalystArtifact` y del mismo a su bucket de Amazon S3. AWS CLI Viene preinstalado y preconfigurado en la plataforma de CodeCatalyst cómputo, por lo que no es necesario instalarlo ni configurarlo.

Para obtener más información sobre el software preempaquetado en la plataforma CodeCatalyst informática, consulte. [Trabajar con imágenes de Docker en un entorno de ejecución](#) Para obtener más información sobre el AWS CLI `aws s3 sync` comando, consulta [sync](#) en la Referencia de AWS CLI comandos.

Para obtener más información sobre la acción de creación, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).

Para crear un flujo de trabajo

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Seleccione Crear flujo de trabajo.
3. Elimina el código de ejemplo de YAML.
4. Agrega el siguiente código YAML:

```
Name: codecatalyst-artifact-workflow
SchemaVersion: 1.0

Triggers:
  - Type: Push
    Branches:
      - main
Actions:
  GenerateFiles:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        # Create the output files.
        - Run: echo "Hello, World!" > "Hello.txt"
        - Run: echo "Goodbye!" > "Goodbye.txt"
```

```

Outputs:
  Artifacts:
    - Name: codecatalystArtifact
      Files:
        - "**/*"
Upload:
  Identifier: aws/build@v1
  DependsOn:
    - GenerateFiles
  Environment:
    Name: codecatalyst-artifact-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-s3-build-role
  Inputs:
    Artifacts:
      - codecatalystArtifact
  Configuration:
    Steps:
      # Upload the output artifact to the S3 bucket.
      - Run: aws s3 sync . s3://codecatalyst-artifact-bucket

```

En el código anterior, sustituya:

- *codecatalyst-artifact-environment* por el nombre del entorno en el que creaste [Requisitos previos](#).
- *codecatalyst-account-connection* con el nombre de la conexión de cuentas en la que creaste [Requisitos previos](#).
- *codecatalyst-s3-build-role* con el nombre de la función de compilación en la que creaste. [Paso 1: Crea un AWS rol](#)
- *codecatalyst-artifact-bucket* con el nombre de la Amazon S3 en la que creaste [Paso 2: Crear un bucket de Amazon S3](#).

Para obtener información sobre las propiedades de este archivo, consulte la [Referencia de acciones de compilación y prueba](#).

5. (Opcional) Selecciona Validar para asegurarte de que el código YAML es válido antes de confirmarlo.
6. Elija Commit (Confirmar).
7. En el cuadro de diálogo Confirmar flujo de trabajo, introduce lo siguiente:

- a. En el nombre del archivo de flujo de trabajo, deje el valor predeterminado, `codecatalyst-artifact-workflow`.
- b. Para el mensaje de confirmación, introduzca:

```
add initial workflow file
```

- c. En Repositorio, elija `codecatalyst-artifact-source-repository`.
- d. En Nombre de sucursal, elija principal.
- e. Elija Commit (Confirmar).

Ahora ha creado un flujo de trabajo. La ejecución de un flujo de trabajo se inicia automáticamente debido al activador definido en la parte superior del flujo de trabajo. En concreto, cuando depositaste (e insertaste) el `codecatalyst-artifact-workflow.yaml` archivo en tu repositorio de origen, el desencadenador inició la ejecución del flujo de trabajo.

Para ver la ejecución del flujo de trabajo en curso

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el flujo de trabajo que acaba de crear: `codecatalyst-artifact-workflow`
3. Elija GenerateFiles ver el progreso de la primera acción de construcción.
4. Selecciona Cargar para ver el progreso de la segunda acción de construcción.
5. Cuando finalice la acción de carga, haz lo siguiente:
 - Si el flujo de trabajo se ejecutó correctamente, continúe con el siguiente procedimiento.
 - Si se produjo un error en la ejecución del flujo de trabajo, seleccione Registros para solucionar el problema.

Paso 5: Verificar los resultados

Cuando se ejecute el flujo de trabajo, vaya al servicio Amazon S3 y busque en su `codecatalyst-artifact-bucket` bucket. Ahora debería incluir los siguientes archivos y carpetas:

```
.  
|- .aws/  
|- .git/
```

```
| Goodbye.txt  
| Hello.txt  
| README.md
```

Los `Hello.txt` archivos `Goodbye.txt` y se cargaron porque formaban parte del `codecatalystArtifact` artefacto. Los `README.md` archivos `.aws/` `.git/`, y se cargaron porque estaban en tu repositorio de origen.

Limpieza

Limpie CodeCatalyst y evite AWS que le cobren por estos servicios.

Para limpiar en CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elimine el repositorio `codecatalyst-artifact-source-repository` de origen.
3. Elimine el `codecatalyst-artifact-workflow` flujo de trabajo.

Para limpiar en AWS

1. Limpie en Amazon S3 de la siguiente manera:
 - a. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
 - b. Elimine los archivos del `codecatalyst-artifact-bucket` depósito.
 - c. Elimine el `codecatalyst-artifact-bucket` depósito.
2. Limpie en IAM de la siguiente manera:
 - a. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - b. Elimine la `codecatalyst-s3-build-policy`.
 - c. Elimine la `codecatalyst-s3-build-role`.

Pruebas mediante flujos de trabajo en CodeCatalyst

En CodeCatalyst, puedes ejecutar pruebas como parte de diferentes acciones del flujo de trabajo, como compilar y probar. Todas estas acciones de flujo de trabajo pueden generar informes. Una acción de prueba es una acción de flujo de trabajo que produce informes de prueba, cobertura de

código, análisis de composición del software y análisis estático. Estos informes se muestran en la CodeCatalyst consola.

Temas

- [Tipos de informes de prueba](#)
- [Añadir la acción de prueba](#)
- [Configuración de informes](#)
- [Uso universal-test-runner](#)
- [Prácticas recomendadas](#)
- [Trabajar con pruebas](#)

Tipos de informes de prueba

La acción CodeCatalyst de prueba de Amazon admite los siguientes tipos de informes. Para ver un ejemplo de cómo dar formato a estos informes en tu YAML, consulta [Ejemplo: configurar informes](#).

Temas

- [Informes de pruebas](#)
- [Informes de cobertura de código](#)
- [Informes de análisis de composición de software](#)
- [Informes de análisis estáticos](#)

Informes de pruebas

En CodeCatalyst, puedes configurar las pruebas unitarias, las pruebas de integración y las pruebas del sistema que se ejecutan durante las compilaciones. A continuación, CodeCatalyst puede crear informes que contengan los resultados de sus pruebas.

Puede usar un informe de prueba para ayudar a solucionar los problemas relacionados con las pruebas. Si tienes muchos informes de pruebas de varias compilaciones, puedes usarlos para ver las tasas de errores y así optimizar tus compilaciones.

Puedes usar los siguientes formatos de archivo de informes de pruebas:

- Cucumber JSON (.json)

- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx, .xml)

Informes de cobertura de código

En CodeCatalyst, puedes generar informes de cobertura de código para tus pruebas. CodeCatalyst proporciona las siguientes métricas de cobertura de código:

Cobertura de línea

Mide el número de afirmaciones que cubren tus pruebas. Una declaración es una sola instrucción que no incluye comentarios.

$$\text{line coverage} = (\text{total lines covered}) / (\text{total number of lines})$$

Cobertura de ramificaciones

Mide cuántas ramas cubren las pruebas de cada rama posible de una estructura de control, como una case sentencia if O.

$$\text{branch coverage} = (\text{total branches covered}) / (\text{total number of branches})$$

Se admiten los siguientes formatos de archivo de informe de cobertura de código:

- JaCoCo XML (.xml)
- SimpleCov [JSON \(generado por simplecov, no por simplecov-json, .json\)](#)
- Clover XML (versión 3, .xml)
- Cobertura XML (.xml)
- LCOV (.info)

Informes de análisis de composición de software

En CodeCatalyst, puede utilizar las herramientas de análisis de composición de software (SCA) para analizar los componentes de su aplicación y comprobar si hay vulnerabilidades de seguridad

conocidas. Puede descubrir y analizar los informes del SARIF que detallan las vulnerabilidades de diferente gravedad y las formas de solucionarlas. Los valores de gravedad válidos, del más grave al menos grave, son:CRITICAL,,, HIGHMEDIUM,LOW. INFORMATIONAL

Se admiten los siguientes formatos de archivo de informes de la SCA:

- SARIF (.sarif, .json)

Informes de análisis estáticos

Puede utilizar los informes de análisis estático (SA) para identificar los defectos del código fuente. En CodeCatalyst, puede generar informes de SA para ayudar a resolver los problemas del código antes de implementarlo. Estos problemas incluyen errores, vulnerabilidades de seguridad, problemas de calidad y otras vulnerabilidades. Los valores de gravedad válidos, del más grave al menos graveCRITICAL, son:HIGH,MEDIUM,LOW, yINFORMATIONAL.

CodeCatalyst proporciona las siguientes métricas de SA:

Errores

Identifica una serie de posibles errores encontrados en el código fuente. Estos errores pueden incluir problemas relacionados con la seguridad de la memoria. El siguiente es un ejemplo de error.

```
// The while loop will inadvertently index into array x out-of-bounds
int x[64];
while (int n = 0; n <= 64; n++) {
    x[n] = 0;
}
```

Vulnerabilidades de

Identifica una serie de posibles vulnerabilidades de seguridad que se encuentran en el código fuente. Estas vulnerabilidades de seguridad pueden incluir problemas como el almacenamiento de los tokens secretos en texto simple.

Problemas de calidad

Identifica una serie de posibles problemas de calidad encontrados en el código fuente. Estos problemas de calidad pueden incluir problemas relacionados con las convenciones de estilo. El siguiente es un ejemplo de un problema de calidad.

```
// The function name doesn't adhere to the style convention of camelCase
int SUBTRACT(int x, int y) {
    return x-y
}
```

Otras vulnerabilidades

Identifica otras posibles vulnerabilidades que se encuentran en el código fuente.

CodeCatalyst admite los siguientes formatos de archivo de informes de SA:

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

Añadir la acción de prueba

Utilice el siguiente procedimiento para añadir una acción de prueba a su flujo de trabajo.

Para añadir una acción de prueba

- Siga las instrucciones en [Añadir la acción de construcción](#). Las instrucciones para añadir una acción de construcción y una acción de prueba son exactamente las mismas.

Definición de acción de prueba

La acción de prueba se define como un conjunto de propiedades YAML dentro del archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte [Referencia de acciones de compilación y prueba](#) en la [Referencia de definición de flujo de trabajo](#).

Configuración de informes

En esta sección se describe cómo configurar un informe de calidad.

Temas

- [Configuración de informes manuales y de detección automática](#)
- [Configurar los criterios de éxito de los informes](#)

- [Ejemplo: configurar informes](#)
- [Soporte SARIF para informes de análisis de composición de software y análisis estático](#)

Configuración de informes manuales y de detección automática

Cuando la detección automática está habilitada, CodeCatalyst busca en todas las entradas transferidas a la acción y en todos los archivos generados por la propia acción, buscando informes de pruebas, cobertura de código, análisis de composición de software (SCA) y análisis estático (SA). Puede ver y manipular cada uno de estos informes en CodeCatalyst.

También puede configurar manualmente los informes que se generan. Puede especificar el tipo de informe que desea generar, así como el formato de archivo. Para obtener más información, consulte [Tipos de informes de prueba](#).

Configurar los criterios de éxito de los informes

Puede establecer los valores que determinan los criterios de éxito de un informe de prueba, cobertura de código, análisis de composición de software (SCA) o análisis estático (SA).

Los criterios de éxito son umbrales que determinan si un informe se aprueba o no. CodeCatalyst primero genera el informe, que puede ser un informe de prueba, de cobertura de código, de SCA o SA, y, a continuación, aplica los criterios de éxito a los informes generados. A continuación, muestra si se cumplieron los criterios de éxito y en qué medida. Si algún informe no cumple los criterios de éxito especificados, se produce un error en la CodeCatalyst acción que los especificó.

Por ejemplo, al establecer los criterios de éxito del informe de la SCA, los valores de vulnerabilidad válidos, que van del más al menos grave, son: CRITICAL,,HIGH, MEDIUMLOW, INFORMATIONAL. Si establece los criterios para buscar una vulnerabilidad en cuanto a su HIGH gravedad, el informe fallará si hay al menos una vulnerabilidad de HIGH gravedad o no hay ninguna vulnerabilidad de HIGH gravedad, pero al menos una vulnerabilidad con un nivel de gravedad superior, como una vulnerabilidad de CRITICAL gravedad superior.

Si no especifica los criterios de conformidad, haga lo siguiente:

- El CodeCatalyst informe que se genere a partir de sus informes sin procesar no mostrará los criterios de éxito.
- Los criterios de éxito no se utilizarán para determinar si la acción del flujo de trabajo asociada se aprueba o no.

Visual

Para configurar los criterios de éxito

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija un flujo de trabajo que contenga una acción que genere un informe. Este es el informe al que desea aplicar los criterios de conformidad. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
3. Elija Editar.
4. Elija Visual.
5. En el diagrama de flujo de trabajo, elija la acción que ha configurado para generar CodeCatalyst informes.
6. Elija la pestaña Salidas.
7. En Detección automática de informes o en Configuración manual de informes, seleccione Criterios de éxito.

Aparecen los criterios de éxito. En función de las selecciones anteriores, es posible que veas alguna de estas opciones o todas ellas:

Tasa de aprobación

Especifique el porcentaje de pruebas en un informe de prueba que deben aprobarse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen números decimales. Por ejemplo: 50, 60.5. Los criterios de porcentaje de aprobados se aplican únicamente a los informes de las pruebas. Para obtener más información sobre los informes de las pruebas, consulte [Informes de pruebas](#).

Cobertura de línea

Especifique el porcentaje de líneas de un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de línea se aplican únicamente a los informes de cobertura de código. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Cobertura de sucursales

Especifique el porcentaje de sucursales en un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de las sucursales se aplican únicamente a los informes de cobertura de códigos. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Vulnerabilidades (SCA)

Especifique el número máximo y la gravedad de las vulnerabilidades permitidas en el informe de la SCA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar las vulnerabilidades, debe especificar:

- La gravedad mínima de las vulnerabilidades que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Por ejemplo, si lo elige HIGH, se HIGH contabilizarán CRITICAL las vulnerabilidades.

- El número máximo de vulnerabilidades de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de vulnerabilidad se aplican únicamente a los informes de la SCA. Para obtener más información sobre los informes de la SCA, consulte. [Informes de análisis de composición de software](#)

Errores

Especifique el número máximo y la gravedad de los errores permitidos en el informe de la SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar los errores, debe especificar:

- La gravedad mínima de los errores que quieres incluir en el recuento. Los valores válidos, del más grave al menos grave, son: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Por ejemplo, si lo eliges HIGH, se HIGH contabilizarán CRITICAL los errores.

- El número máximo de errores de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de errores se aplican únicamente a PyLint los informes de ESLint SA. Para obtener más información sobre los informes de SA, consulte. [Informes de análisis estáticos](#)

Vulnerabilidades de

Especifique el número máximo y la gravedad de las vulnerabilidades de seguridad permitidas en el informe de la SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar las vulnerabilidades de seguridad, debe especificar:

- La gravedad mínima de las vulnerabilidades de seguridad que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo deseaHIGH, se HIGH contabilizarán las vulnerabilidades de CRITICAL seguridad.

- El número máximo de vulnerabilidades de seguridad de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de vulnerabilidad de seguridad se aplican únicamente a PyLint los informes de ESLint SA. Para obtener más información sobre los informes de SA, consulte. [Informes de análisis estáticos](#)

Problemas de calidad

Especifique el número máximo y la gravedad de los problemas de calidad permitidos en el informe de la SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar los problemas de calidad, debe especificar:

- La gravedad mínima de los problemas de calidad que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo deseaHIGH, se HIGH contabilizarán los problemas de CRITICAL calidad.

- El número máximo de problemas de calidad de la gravedad especificada que desee permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de calidad solo se aplican a los informes PyLint de ESLint SA. Para obtener más información sobre los informes de SA, consulte [Informes de análisis estáticos](#)

8. Elija Commit (Confirmar).
9. Ejecute su flujo de trabajo para CodeCatalyst aplicar los criterios de éxito a sus informes sin procesar y regenere los CodeCatalyst informes asociados con la información sobre los criterios de éxito incluida. Para obtener más información, consulte [Iniciar una ejecución de flujo de trabajo](#).

YAML

Para configurar los criterios de éxito

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija un flujo de trabajo que contenga una acción que genere un informe. Este es el informe al que desea aplicar los criterios de conformidad. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
3. Elija Editar.
4. Elige YAML.
5. En el diagrama de flujo de trabajo, elija la acción que ha configurado para generar CodeCatalyst informes.
6. En el panel de detalles, seleccione la pestaña Salidas.
7. En la acción, en la `AutoDiscoverReports` sección o en la `Reports` sección, añada una `SuccessCriteria` propiedad junto con `PassRate`, `LineCoverage`, `BranchCoverage`, `Vulnerabilities`, `StaticAnalysisBug`, `StaticAnalysisS` y `StaticAnalysisQuality` propiedades.

Para obtener una explicación de cada una de estas propiedades, consulte la [Referencia de acciones de compilación y prueba](#).

8. Elija Commit (Confirmar).

9. Ejecute su flujo de trabajo para CodeCatalyst aplicar los criterios de éxito a sus informes sin procesar y regenere los CodeCatalyst informes asociados con la información sobre los criterios de éxito incluida. Para obtener más información sobre cómo iniciar un flujo de trabajo, consulte [Iniciar una ejecución de flujo de trabajo](#).

Ejemplo: configurar informes

El siguiente ejemplo muestra cómo configurar manualmente cuatro informes: un informe de prueba, un informe de cobertura de código, un informe de análisis de la composición del software y un informe de análisis estático.

```
Reports:
  MyTestReport:
    Format: JUNITXML
    IncludePaths:
      - "*.xml"
    ExcludePaths:
      - report1.xml
    SuccessCriteria:
      PassRate: 90
  MyCoverageReport:
    Format: CLOVERXML
    IncludePaths:
      - output/coverage/jest/clover.xml
    SuccessCriteria:
      LineCoverage: 75
      BranchCoverage: 75
  MySCAReport:
    Format: SARIFSCA
    IncludePaths:
      - output/sca/reports.xml
    SuccessCriteria:
      Vulnerabilities:
        Number: 5
        Severity: HIGH
  MySAReport:
    Format: ESLINTJSON
    IncludePaths:
      - output/static/eslint.xml
    SuccessCriteria:
      StaticAnalysisBug:
        Number: 10
```



```
Severity: MEDIUM
StaticAnalysisSecurity:
  Number: 5
  Severity: CRITICAL
StaticAnalysisQuality:
  Number: 0
  Severity: INFORMATIONAL
```

Soporte SARIF para informes de análisis de composición de software y análisis estático

El formato SARIF (Static Analysis Results Interchange Format) es un formato de archivo de salida que está disponible en los informes de análisis de composición de software y análisis estático. CodeCatalyst El siguiente ejemplo muestra cómo configurar manualmente el SARIF en un informe de análisis estático:

```
Reports:
  MySAReport:
    Format: SARIFSA
    IncludePaths:
      - output/sa_report.json
    SuccessCriteria:
      StaticAnalysisFinding:
        Number: 25
        Severity: HIGH
```

CodeCatalyst admite las siguientes propiedades de SARIF, que se pueden utilizar para optimizar la forma en que aparecerán los resultados del análisis en sus informes.

Temas

- [Objeto sarifLog](#)
- [Objeto run](#)
- [Objeto toolComponent](#)
- [Objeto reportingDescriptor](#)
- [Objeto result](#)
- [Objeto location](#)
- [Objeto physicalLocation](#)
- [Objeto logicalLocation](#)

- [Objeto fix](#)

Objeto **sarifLog**

Nombre	Obligatoria	Descripción
<code>\$schema</code>	Sí	El URI del esquema JSON de SARIF para la versión 2.1.0.
<code>version</code>	Sí	CodeCatalyst solo es compatible con la versión 2.1.0 de SARIF.
<code>runs[]</code>	Sí	Un archivo SARIF contiene una matriz de una o más ejecuciones, cada una de las cuales representa una sola ejecución de la herramienta de análisis.

Objeto **run**

Nombre	Obligatoria	Descripción
<code>tool.driver</code>	Sí	<code>toolComponent</code> Objeto que describe la herramienta de análisis.
<code>tool.name</code>	No	Propiedad que indica el nombre de la herramienta utilizada para realizar el análisis.
<code>results[]</code>	Sí	Los resultados de la herramienta de análisis que se muestran en CodeCatalyst.

Objeto **toolComponent**

Nombre	Obligatoria	Descripción
<code>name</code>	Sí	El nombre de la herramienta de análisis.
<code>properties.artifactScanned</code>	No	Número total de artefactos analizados por la herramienta.
<code>rules[]</code>	Sí	Conjunto de <code>reportingDescriptor</code> objetos que representan reglas. Según estas reglas, la herramienta de análisis encuentra problemas en el código que se analiza.

Objeto **reportingDescriptor**

Nombre	Obligatoria	Descripción
<code>id</code>	Sí	El identificador único de la regla que se utiliza para hacer referencia a un hallazgo. Longitud máxima: 1024 caracteres
<code>name</code>	No	El nombre para mostrar de la regla. Longitud máxima: 1024 caracteres
<code>shortDescription.text</code>	No	Descripción abreviada de la regla.

Nombre	Obligatoria	Descripción
		Longitud máxima: 3000 caracteres
<code>fullDescription.text</code>	No	Una descripción completa de la regla. Longitud máxima: 3000 caracteres
<code>helpUri</code>	No	Una cadena que se puede localizar para que contenga el URI absoluto de la documentación principal de la regla. Longitud máxima: 3000 caracteres
<code>properties.unscore</code>	No	Una bandera que indica si se ha puntuado el resultado del escaneo.
<code>properties.score.severity</code>	No	Un conjunto fijo de cadenas que especifican el nivel de gravedad del hallazgo. Longitud máxima: 1024 caracteres
<code>properties.cvssv3_baseSeverity</code>	No	Una calificación cualitativa de gravedad del Common Vulnerability Scoring System v3.1 .
<code>properties.cvssv3_baseScore</code>	No	Una puntuación base del CVSS v3 que oscila entre 0.0 y 10.0.

Nombre	Obligatoria	Descripción
<code>properties.cvssv2_severity</code>	No	Si los valores de CVSS v3 no están disponibles, CodeCatalyst busca los valores de CVSS v2.
<code>properties.cvssv2_score</code>	No	Una puntuación base del CVSS v2 que oscila entre 0.0 y 10.0.
<code>properties.severity</code>	No	Un conjunto fijo de cadenas que especifican el nivel de gravedad del hallazgo. Longitud máxima: 1024 caracteres
<code>defaultConfiguration.level</code>	No	La severidad predeterminada de una regla.

Objeto **result**

Nombre	Obligatoria	Descripción
<code>ruleId</code>	Sí	El identificador único de la regla que se utiliza para hacer referencia a un hallazgo. Longitud máxima: 1024 caracteres
<code>ruleIndex</code>	Sí	El índice de la regla asociada en el componente <code>rules[]</code> de la herramienta.

Nombre	Obligatoria	Descripción
<code>message.text</code>	Sí	Un mensaje que describe el resultado y muestra el mensaje de cada hallazgo. Longitud máxima: 3000 caracteres
<code>rank</code>	No	Un valor entre 0.0 y 100.0, ambos inclusive, que representa la prioridad o la importancia del resultado. Esta escala valora 0.0 como la prioridad más baja y 100,0 como la prioridad más alta.
<code>level</code>	No	La gravedad del resultado. Longitud máxima: 1024 caracteres
<code>properties.unscore</code>	No	Una bandera que indica si se ha puntuado el resultado del escaneo.
<code>properties.score.severity</code>	No	Un conjunto fijo de cadenas que especifican el nivel de gravedad del hallazgo. Longitud máxima: 1024 caracteres
<code>properties.cvssv3_baseSeverity</code>	No	Una calificación cualitativa de gravedad del Common Vulnerability Scoring System v3.1 .

Nombre	Obligatoria	Descripción
<code>properties.cvssv3_baseScore</code>	No	Una puntuación base del CVSS v3 que oscila entre 0.0 y 10.0.
<code>properties.cvssv2_severity</code>	No	Si los valores de CVSS v3 no están disponibles, CodeCatalyst busca los valores de CVSS v2.
<code>properties.cvssv2_score</code>	No	Una puntuación base del CVSS v2 que oscila entre 0.0 y 10.0.
<code>properties.severity</code>	No	Un conjunto fijo de cadenas que especifican el nivel de gravedad del hallazgo. Longitud máxima: 1024 caracteres
<code>locations[]</code>	Sí	El conjunto de ubicaciones en las que se detectó el resultado . Solo se debe incluir una ubicación, a menos que el problema solo pueda corregirse realizando un cambio en cada ubicación especificada. CodeCatalyst utiliza el primer valor de la matriz de ubicación para anotar el resultado. Número máximo de location objetos: 10

Nombre	Obligatoria	Descripción
<code>relatedLocations[]</code>	No	Una lista de ubicaciones adicionales a las que se hace referencia en el hallazgo. Número máximo de <code>location</code> objetos: 50
<code>fixes[]</code>	No	Conjunto de <code>fix</code> objetos que representan la recomendación proporcionada por la herramienta de digitalización. CodeCatalyst utiliza la primera recomendación de la <code>fixes</code> matriz.

Objeto **location**

Nombre	Obligatoria	Descripción
<code>physicalLocation</code>	Sí	Identifica el artefacto y la región.
<code>logicalLocations[]</code>	No	El conjunto de ubicaciones descritas por su nombre sin hacer referencia al artefacto.

Objeto **physicalLocation**

Nombre	Obligatoria	Descripción
<code>artifactLocation.uri</code>	Sí	El URI que indica la ubicación de un artefacto, normalmente un archivo en el repositor

Nombre	Obligatoria	Descripción
		io o generado durante una compilación.
<code>fileLocation.uri</code>	No	El URI alternativo que indica la ubicación del archivo. Se usa si se <code>artifactLocation.uri</code> devuelve vacío.
<code>region.startLine</code>	Sí	El número de línea del primer carácter de la región.
<code>region.startColumn</code>	Sí	El número de columna del primer carácter de la región.
<code>region.endLine</code>	Sí	El número de línea del último carácter de la región.
<code>region.endColumn</code>	Sí	El número de columna del último carácter de la región.

Objeto **logicalLocation**

Nombre	Obligatoria	Descripción
<code>fullyQualifiedname</code>	No	Información adicional que describe la ubicación del resultado. Longitud máxima: 1024 caracteres

Objeto **fix**

Nombre	Obligatoria	Descripción
<code>description.text</code>	No	Un mensaje que muestra una recomendación para cada hallazgo. Longitud máxima: 3000 caracteres
<code>artifactChanges.[0].artifactLocation.uri</code>	No	El URI que indica la ubicación del artefacto que debe actualizarse.

Uso universal-test-runner

Las acciones de prueba se integran con la herramienta `universal-test-runner` de línea de comandos de código abierto. Esta herramienta proporciona funciones de prueba avanzadas, como volver a intentar uno o más casos de prueba a partir de un informe de prueba. `universal-test-runner` utiliza el [protocolo de ejecución](#) de pruebas para ejecutar las pruebas en cualquier idioma de un marco determinado. `universal-test-runner` es compatible con los siguientes marcos:

- [Gradle](#)
- [¿Es broma?](#)
- [Maven](#)
- [más puñetero](#)
- [.NET](#)

`universal-test-runner` se instala solo en las imágenes seleccionadas para realizar acciones de prueba. Si configura una acción de prueba para usar un Docker Hub o Amazon ECR personalizados, debe instalarla manualmente `universal-test-runner` para habilitar las funciones de prueba avanzadas. Para ello, instale Node.js (14 o superior) en la imagen y, a continuación, instálelo `universal-test-runner` npm mediante el comando shell. - Run: `npm install -g @aws/universal-test-runner` Para obtener más información sobre cómo instalar Node.js en el

contenedor mediante comandos de shell, consulte [Instalación y actualización de Node Version Manager](#).

Para obtener más información al respecto `universal-test-runner`, consulte [¿Qué es universal-test-runner?](#)

Visual

Para usar `universal-test-runner` en el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo.
4. Elija Editar.
5. Elige Visual.
6. Elija Acciones.
7. En Acciones, selecciona Probar.
8. En la pestaña Configuración, complete el campo de comandos de Shell actualizando el código de ejemplo con los marcos compatibles que haya elegido. Por ejemplo, para usar un marco compatible, usaría un Run comando similar al siguiente.

```
- Run: run-tests <framework>
```

Si el marco que desea no es compatible, considere la posibilidad de añadir un adaptador o un ejecutor personalizados. Para obtener una descripción del campo de comandos de Shell, consulte [Steps](#).

9. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para usar `universal-test-runner` en el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo.
4. Elija Editar.
5. Elige YAML.
6. Elija Acciones.
7. En Acciones, selecciona Probar.
8. Modifica el código YAML según tus necesidades. Por ejemplo, para usar un marco compatible, usaría un Run comando similar al siguiente.

```
Configuration:
```

```
Steps:
```

```
- Run: run-tests <framework>
```

Si el marco que desea no es compatible, considere la posibilidad de añadir un adaptador o un ejecutor personalizados. Para obtener una descripción de la propiedad Steps, consulte [Steps](#).

9. (Opcional) Elija Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Prácticas recomendadas

Cuando utilice las funciones de prueba que proporciona CodeCatalyst, le recomendamos que siga estas prácticas recomendadas.

Temas

- [Descubrimiento automático](#)
- [Criterios correctos](#)
- [Incluir/excluir rutas](#)

Descubrimiento automático

Al configurar las acciones en CodeCatalyst, la detección automática le permite descubrir automáticamente los resultados de varias herramientas, como los informes de pruebas de JUnit,

y generar CodeCatalyst informes relevantes a partir de ellos. La detección automática ayuda a garantizar que los informes se sigan generando aunque cambien los nombres o las rutas hacia los resultados descubiertos. Cuando se añaden nuevos archivos, los descubre CodeCatalyst automáticamente y genera los informes pertinentes. Sin embargo, si utilizas la detección automática, es importante tener en cuenta algunos de los siguientes aspectos de esta función:

- Cuando activas la detección automática en tu acción, todos los informes del mismo tipo detectados automáticamente compartirán los mismos criterios de éxito. Por ejemplo, un criterio compartido, como la tasa mínima de aprobación, se aplicaría a todos los informes de pruebas detectados automáticamente. Si necesita criterios diferentes para los informes del mismo tipo, debe configurar cada uno de estos informes de forma explícita.
- La detección automática también puede encontrar los informes generados por sus dependencias y, si se configuran los criterios de éxito, es posible que no se realice ninguna acción en estos informes. Este problema se puede solucionar actualizando la configuración de la ruta de exclusión.
- No se garantiza que la detección automática produzca siempre la misma lista de informes, ya que analiza la acción en tiempo de ejecución. En el caso de que desee que siempre se genere un informe en particular, debe configurar los informes de forma explícita. Por ejemplo, si las pruebas dejaran de ejecutarse como parte de la compilación, el marco de pruebas no generaría ningún resultado y, en consecuencia, no se generaría ningún informe de prueba y la acción podría tener éxito. Si quieres que el éxito de tu acción dependa de esa prueba en particular, debes configurar ese informe de forma explícita.

Tip

Al comenzar con un proyecto nuevo o existente, usa la detección automática para todo el directorio del proyecto (incluir**/*). Esto implica la generación de informes en todos los archivos del proyecto, incluidos los que se encuentran en los subdirectorios.

Para obtener más información, consulte [Configuración de informes](#).

Criterios correctos

Puede imponer umbrales de calidad en sus informes configurando los criterios de éxito. Por ejemplo, si se detectaron automáticamente dos informes de cobertura de código, uno con una cobertura de línea del 80% y el otro con una cobertura de línea del 60%, tiene las siguientes opciones:

- Establezca los criterios de éxito de la detección automática para la cobertura de la línea en un 80%. Esto provocaría que se aprobara el primer informe y que el segundo no, lo que provocaría el fracaso de la acción general. Para desbloquear el flujo de trabajo, añada nuevas pruebas a su proyecto hasta que la cobertura de líneas del segundo informe supere el 80%.
- Establezca los criterios de éxito de la detección automática para la cobertura de la línea en un 60%. Esto haría que ambos informes se aprobaran, lo que haría que la acción se llevara a cabo correctamente. A continuación, podría trabajar para aumentar la cobertura del código en el segundo informe. Sin embargo, con este enfoque, no se puede garantizar que la cobertura del primer informe no caiga por debajo del 80%.
- Configura de forma explícita uno o ambos informes mediante el editor visual o añade una sección y una ruta de YAML explícitas para cada informe. Esto le permitiría configurar criterios de éxito independientes y nombres personalizados para cada informe. Sin embargo, con este enfoque, la acción podría fallar si las rutas de los informes cambian.

Para obtener más información, consulte [Configurar los criterios de éxito de los informes](#).

Incluir/excluir rutas

Al revisar los resultados de las acciones, puede ajustar la lista de informes que se generan CodeCatalyst `IncludePaths` configurando `y.ExcludePaths`

- `IncludePaths` Utilícelo para especificar los archivos y las rutas de archivo CodeCatalyst que desea incluir al buscar informes. Por ejemplo, si lo especifica `"/test/report/*"`, CodeCatalyst busca en el `/test/report/` directorio toda la imagen de creación utilizada por la acción. Cuando encuentra ese directorio, CodeCatalyst busca los informes en ese directorio.

Note

En el caso de los informes configurados manualmente, `IncludePaths` debe ser un patrón global que coincida con un único archivo.

- `ExcludePaths` Utilícelo para especificar los archivos y las rutas de archivo CodeCatalyst que desea excluir al buscar informes. Por ejemplo, si lo especifica `"/test/reports/**/*"`, no CodeCatalyst buscará archivos en el `/test/reports/` directorio. Para ignorar todos los archivos de un directorio, utilice el patrón `**/*` global.

Los siguientes son ejemplos de posibles patrones globales.

Patrón	Descripción
<code>*.*</code>	Coincide con todos los nombres de objetos del directorio actual que contienen un punto
<code>*.xml</code>	Coincide con todos los nombres de objetos del directorio actual que terminan en <code>.xml</code>
<code>*.{xml,txt}</code>	Coincide con todos los nombres de objetos del directorio actual que terminan en <code>.xml</code> o <code>.txt</code>
<code>**/*.xml</code>	Coincide con los nombres de los objetos de todos los directorios que terminan en <code>.xml</code>
<code>testFolder</code>	Coincide con un objeto llamado <code>testFolder</code> y lo trata como un archivo
<code>testFolder/*</code>	Coincide con los objetos de un nivel de la subcarpeta de <code>testFolder</code> origen, como <code>testFolder/file.xml</code>
<code>testFolder/**</code>	Hace coincidir los objetos de dos niveles de la subcarpeta de <code>testFolder</code> , como <code>testFolder/reportsFolder/file.xml</code>
<code>testFolder/**</code>	Coincide con la subcarpeta <code>testFolder</code> así como con los archivos debajo de <code>testFolder</code> , como <code>testFolder/file.xml</code> y <code>testFolder/otherFolder/file.xml</code>

CodeCatalyst interpreta los patrones globales de la siguiente manera:

- El carácter de barra diagonal (`/`) separa los directorios en las rutas de los archivos.
- 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.

Note

ExcludePath tiene prioridad sobre IncludePaths. Si ambos ExcludePaths incluyen IncludePaths la misma carpeta, esa carpeta no se escanea en busca de informes.

Trabajar con pruebas

En esta sección se describe cómo ver y configurar las acciones de prueba.

Temas

- [Visualización de los resultados de una acción de prueba](#)
- [Omitir las pruebas fallidas](#)
- [Reintentar casos de prueba](#)

Visualización de los resultados de una acción de prueba

Siga las instrucciones siguientes para ver los resultados de una acción de prueba, incluidos los registros, los informes y las variables generados.

Para ver los resultados de una acción de prueba

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
3. En el diagrama de flujo de trabajo, elige el nombre de la acción de prueba (por ejemplo, Probar).
4. Para ver los registros generados por una acción, selecciona Registros. Se muestran los registros de las distintas fases de la acción. Puede expandir o contraer los registros según sea necesario.
5. Para ver los informes de prueba generados por la acción de prueba, seleccione Informes. Para obtener más información, consulte [Tipos de informes de prueba](#).
6. Para ver la configuración utilizada para la acción de prueba, elija Configuración. Para obtener más información, consulte [Añadir la acción de prueba](#).

7. Para ver las variables utilizadas por la acción de prueba, elija Variables. Para obtener más información, consulte [Trabajar con variables](#).

Omitir las pruebas fallidas

Si su acción tiene más de un comando de prueba, puede que desee permitir que los comandos de prueba posteriores de la acción se ejecuten incluso si se produce un error en uno de los comandos anteriores. Por ejemplo, en los siguientes comandos, puede que desee `test2` ejecutarlos siempre, incluso si `test1` se produce un error.

Steps:

- Run: `npm install`
- Run: `npm run test1`
- Run: `npm run test2`

Normalmente, cuando un paso devuelve un error, Amazon CodeCatalyst detiene la acción del flujo de trabajo y la marca como fallida. Puedes permitir que los pasos de la acción continúen ejecutándose redirigiendo el resultado del error `null`. Para ello, añade `2>/dev/null` al comando. Con esta modificación, el ejemplo anterior tendría el aspecto siguiente.

Steps:

- Run: `npm install`
- Run: `npm run test1 2>/dev/null`
- Run: `npm run test2`

En el segundo fragmento de código, se respetará el estado del `npm install` comando, pero se ignorará cualquier error que devuelva el `npm run test1` comando. Como resultado, se ejecuta el `npm run test2` comando. De este modo, podrá ver ambos informes a la vez, independientemente de si se produce un error.

Reintentar casos de prueba

Si el informe falla debido a varios casos de prueba, solo puede volver a intentar esas pruebas individuales. Esto le permite comprobar rápidamente la calidad de los casos de prueba y determinar los siguientes pasos para resolver los problemas, como eliminar una dependencia interrumpida o iniciar una nueva ejecución del flujo de trabajo. La acción de prueba incorpora la opción de volver `universal-test-runner` a intentar solo los casos de prueba seleccionados, en lugar de toda la acción. Solo puedes volver a intentar un conjunto de casos de prueba seleccionados por acción

a la vez y solo puedes volver a intentarlo cinco veces por informe de prueba. Para obtener más información, consulte [Uso universal-test-runner](#).

Note

Si vuelve a intentar los casos de prueba en un informe, no afectará al estado del flujo de trabajo que generó el informe original.

Siga las instrucciones siguientes para volver a intentar los casos de prueba de sus informes.

Para volver a intentar los casos de prueba de un informe

1. En el panel de navegación, elija Reports.
2. Elija el nombre del informe. Puede filtrar por nombre, estado, repositorio, sucursal o tipo de informe.
3. En el nombre del informe, elija Resultados.
4. Seleccione los casos de prueba que desee volver a intentar, elija Volver a ejecutar y, a continuación, elija Casos de prueba seleccionados.
5. Cuando finalice el reintento, selecciona Actualizar en el encabezado y consulta los resultados actualizados.

Implementación mediante flujos de trabajo en CodeCatalyst

Con [CodeCatalyst los flujos de trabajo](#), puede implementar aplicaciones y otros recursos en varios objetivos AWS Lambda, como Amazon ECS y más.

Temas

- [¿Cómo puedo implementar una aplicación?](#)
- [Lista de acciones de despliegue](#)
- [Ventajas de las acciones de despliegue](#)
- [Alternativas para implementar acciones](#)
- [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#)
- [Tutorial: Implementación de una aplicación en Amazon ECS](#)

- [Tutorial: Implementación de una aplicación en Amazon EKS](#)
- [Añadir la acción «Desplegar AWS CloudFormation pila»](#)
- [Añadir la acción «Implementar en Amazon ECS»](#)
- [Añadir la acción «Implementar en un clúster de Kubernetes»](#)
- [Añadir la acción «AWS CDK desplegar»](#)
- [Trabajo con implementaciones](#)

¿Cómo puedo implementar una aplicación?

Para implementar una aplicación o un recurso CodeCatalyst, primero debe crear un flujo de trabajo y, a continuación, especificar una acción de implementación dentro del mismo. Una acción de despliegue es un componente básico del flujo de trabajo que define qué es lo que se quiere implementar, dónde se quiere implementar y cómo se quiere implementar (por ejemplo, mediante un esquema azul/verde). Puedes añadir una acción de despliegue a tu flujo de trabajo mediante el editor visual de la CodeCatalyst consola o el editor YAML.

Los pasos generales para implementar una aplicación o un recurso son los siguientes.

Para implementar una aplicación (tareas de alto nivel)

1. En tu CodeCatalyst proyecto, agregas el código fuente de la aplicación que deseas implementar. Para obtener más información, consulte [Trabajar con repositorios de código fuente en CodeCatalyst](#).
2. En tu CodeCatalyst proyecto, creas un flujo de trabajo. En el flujo de trabajo se define cómo crear, probar e implementar la aplicación. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).
3. En el flujo de trabajo, se añade un disparador, una acción de creación y, de forma opcional, una acción de prueba. Para obtener más información, consulte [Trabajar con factores desencadenantes](#), [Añadir la acción de construcción](#) y [Añadir la acción de prueba](#).
4. En el flujo de trabajo, se añade una acción de despliegue. Puede elegir entre varias acciones de implementación CodeCatalyst proporcionadas para su aplicación en diferentes destinos, como Amazon ECS. (También puede usar una acción de compilación o una GitHub acción para implementar la aplicación. Para obtener más información sobre la acción de creación y GitHub las acciones, consulte [Alternativas para implementar acciones](#).)
5. El flujo de trabajo se inicia de forma manual o automática mediante un disparador. El flujo de trabajo ejecuta las acciones de creación, prueba e implementación en secuencia para

implementar la aplicación y los recursos en el destino. Para obtener más información, consulte [Iniciar una ejecución de flujo de trabajo](#).

Lista de acciones de despliegue

Están disponibles las siguientes acciones de despliegue:

- Implementar AWS CloudFormation pila: esta acción crea una CloudFormation pila en AWS función de una [AWS CloudFormation plantilla](#) o [AWS Serverless Application Model plantilla](#) que proporciones. Para obtener más información, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).
- Implementar en Amazon ECS: esta acción registra un archivo de [definición de tareas](#) que usted proporciona. Para obtener más información, consulte [Añadir la acción «Implementar en Amazon ECS»](#).
- Implementar en un clúster de Kubernetes: esta acción implementa una aplicación en un clúster de Amazon Elastic Kubernetes Service. Para obtener más información, consulte [Añadir la acción «Implementar en un clúster de Kubernetes»](#).
- AWS CDK [implementar](#): esta acción implementa una aplicación en AWS CDK AWS Para obtener más información, consulte [Añadir la acción «AWS CDK desplegar»](#).

Note

Existen otras CodeCatalyst acciones que pueden implementar recursos; sin embargo, no se consideran acciones de implementación porque su información de implementación no aparece en la página Entornos. Para obtener más información sobre la página de entornos y sobre la visualización de las implementaciones, consulte [Trabajo con entornos](#) y [Trabajo con implementaciones](#).

Ventajas de las acciones de despliegue

El uso de acciones de despliegue dentro de un flujo de trabajo tiene las siguientes ventajas:

- Historial de despliegues: consulta un historial de tus despliegues para ayudarte a gestionar y comunicar los cambios en el software implementado.

- Trazabilidad: realice un seguimiento del estado de sus despliegues a través de la CodeCatalyst consola y compruebe cuándo y dónde se implementó cada revisión de la aplicación.
- Reversiones: revierte las implementaciones automáticamente si hay errores. También puede configurar alarmas para activar las reversiones de la implementación.
- Supervisión: observe su implementación a medida que avanza en las distintas etapas de su flujo de trabajo.
- Integración con otras CodeCatalyst funciones: almacene el código fuente y, a continuación, compruébelo, pruébelo e impleméntelo, todo desde una sola aplicación.

Alternativas para implementar acciones

No es necesario utilizar las acciones de despliegue, aunque se recomiendan porque ofrecen las ventajas descritas en la sección anterior. En su lugar, puede utilizar las siguientes [CodeCatalyst acciones](#):

- Una acción de construcción.

Por lo general, las acciones de creación se utilizan si se quiere realizar el despliegue en un destino para el que no existe la acción de despliegue correspondiente o si se desea tener más control sobre el procedimiento de despliegue. Para obtener más información sobre el uso de acciones de compilación para implementar recursos, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).

- Una GitHub acción.

Puede usar una [GitHub acción](#) dentro de un CodeCatalyst flujo de trabajo para implementar aplicaciones y recursos (en lugar de una CodeCatalyst acción). Para obtener información sobre cómo utilizar GitHub las acciones dentro de un CodeCatalyst flujo de trabajo, consulte [Añadir GitHub acciones](#)

También puedes usar los siguientes AWS servicios para implementar tu aplicación, si no quieres usar un CodeCatalyst flujo de trabajo para hacerlo:

- AWS CodeDeploy — consulte [¿Qué es CodeDeploy?](#)
- AWS CodeBuild y AWS CodePipeline — consulte [¿Qué es AWS CodeBuild?](#) y [¿Qué es AWS CodePipeline?](#)
- AWS CloudFormation — ver [¿Qué es AWS CloudFormation?](#)

Uso CodeDeploy CodeBuild y CodePipeline CloudFormation servicios para despliegues empresariales complejos.

Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation

En este tutorial, aprenderá a crear, probar e implementar una aplicación sin servidor mediante un flujo de trabajo y. AWS CloudFormation

La aplicación de este tutorial es una aplicación web sencilla que genera un mensaje de «Hola mundo». Se compone de una AWS Lambda función y un Amazon API Gateway, y se crea mediante [AWS Serverless Application Model \(AWS SAM\)](#), que es una extensión de [AWS CloudFormation](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Crea un repositorio de código fuente](#)
- [Paso 2: Crear AWS roles](#)
- [Paso 3: Añada AWS funciones a CodeCatalyst](#)
- [Paso 4: Crear un bucket de Amazon S3](#)
- [Paso 5: Añadir los archivos fuente](#)
- [Paso 6: Crear y ejecutar un flujo de trabajo](#)
- [Paso 7: Realizar un cambio](#)
- [Limpieza](#)

Requisitos previos

Antes de empezar

- Necesita un CodeCatalyst espacio con una AWS cuenta conectada. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- En tu espacio, necesitas un CodeCatalyst proyecto vacío para empezar desde cero llamado:

```
codecatalyst-cfn-project
```

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

- En tu proyecto, necesitas un CodeCatalyst entorno llamado:

```
codecatalyst-cfn-environment
```

Configure este entorno de la siguiente manera:

- Elija cualquier tipo, como el de no producción.
- Conecta tu AWS cuenta a ella.

Para obtener más información, consulte [Trabajo con entornos](#).

Paso 1: Crea un repositorio de código fuente

En este paso, creará un repositorio de fuentes en CodeCatalyst. Este repositorio se utiliza para almacenar los archivos fuente del tutorial, como el archivo de funciones Lambda.

Para obtener más información sobre los repositorios de fuentes, consulte [Crear un repositorio de código fuente](#)

Para crear un repositorio de origen

1. En CodeCatalyst, en el panel de navegación, elija Código y, a continuación, elija Repositorios de código fuente.
2. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
3. En Nombre del repositorio, introduzca:

```
codecatalyst-cfn-source-repository
```

4. Seleccione Crear.

Ahora ha creado un repositorio llamado `codecatalyst-cfn-source-repository`.

Paso 2: Crear AWS roles

En este paso, creará las siguientes funciones de AWS IAM:

- Función de implementación: otorga a la acción CodeCatalyst Deploy AWS CloudFormation stack el permiso para acceder a la AWS cuenta y al CloudFormation servicio en los que se implementará la aplicación sin servidor. La acción Implementar AWS CloudFormation pila forma parte de tu flujo de trabajo.

- Función de creación: concede a la acción de CodeCatalyst creación el permiso para acceder a su AWS cuenta y escribir en Amazon S3, donde se almacenará el paquete de aplicaciones sin servidor. La acción de creación forma parte de su flujo de trabajo.
- Función de pila: otorga CloudFormation permiso para leer y modificar los recursos especificados en la AWS SAM plantilla que proporcionará más adelante. También concede permiso a CloudWatch.

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

Note

Para ahorrar tiempo, puede crear un único rol, denominado `CodeCatalystWorkflowDevelopmentRole-spaceName` rol, en lugar de los tres roles enumerados anteriormente. Para obtener más información, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Tenga en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios que pueden suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor. En este tutorial se da por sentado que está creando los tres roles enumerados anteriormente.

Note

También se requiere un [rol de ejecución de Lambda](#), pero no es necesario que lo cree ahora porque el `sam-template.yml` archivo lo crea automáticamente cuando ejecuta el flujo de trabajo en el paso 5.

Para crear un rol de despliegue

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

- c. En el panel de navegación, seleccione Políticas.
- d. Elija Crear política.
- e. Seleccione la pestaña JSON.
- f. Elimina el código existente.
- g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

codecatalyst-deploy-policy

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de implementación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. En Políticas de permisos, busque `codecatalyst-deploy-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-deploy-role

- i. En la descripción del rol, introduzca:

CodeCatalyst deploy role

- j. Seleccione Crear rol.

Ahora ha creado un rol de despliegue con una política de confianza y una política de permisos.

3. Obtenga el ARN del rol de implementación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (codecatalyst-deploy-role).
 - c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de implementación con los permisos adecuados y ha obtenido su ARN.

Para crear un rol de compilación

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Elija Crear política.
 - e. Seleccione la pestaña JSON.
 - f. Elimina el código existente.
 - g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ]
  }]
}
```

```

    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-build-policy
```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [

```

```
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
    ],
    },
    "Action": "sts:AssumeRole"
}
]
```

- e. Elija Siguiente.
- f. En Políticas de permisos, busque `codecatalyst-build-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-build-role

- i. En la descripción del rol, introduzca:

CodeCatalyst build role

- j. Seleccione Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

3. Obtenga el ARN del rol de compilación de la siguiente manera:

- a. Seleccione Roles en el panel de navegación.
- b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-build-role`).
- c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de compilación con los permisos adecuados y ha obtenido su ARN.

Para crear un rol de pila

1. Inicia sesión AWS con la cuenta en la que quieres implementar tu stack.
2. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
3. Cree el rol de pila de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. Elija Crear rol.
 - c. Elija Servicio de AWS .
 - d. En la sección Caso de uso, elija una opción CloudFormation de la lista desplegable.
 - e. Seleccione el botón de opción CloudFormation.
 - f. En la parte inferior, selecciona Siguiente.
 - g. En el cuadro de búsqueda, busque las siguientes políticas de permisos y, a continuación, active las casillas correspondientes.

Note

Si buscas una política y no aparece, asegúrate de seleccionar Borrar filtros e inténtalo de nuevo.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- YO SOY FullAccess
- AWS Lambda_ FullAccess
- API de Amazon GatewayAdministrator
- Amazon S3 FullAccess
- Amazon EC2 ContainerRegistryFullAccess

La primera política permite el acceso CloudWatch a para permitir la reversión de las pilas cuando se produce una alarma.

Las políticas restantes permiten acceder AWS SAM a los servicios y recursos de la pila que se implementarán en este tutorial. Para obtener más información, consulte [los permisos](#) en la [Guía para AWS Serverless Application Model desarrolladores](#).

- h. Elija Siguiente.
 - i. En el nombre del rol, escriba:

`codecatalyst-stack-role`
 - j. Seleccione Crear rol.
4. Obtenga el ARN del rol de pila, de la siguiente manera:
- a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-stack-role`).
 - c. Elija el rol de la lista.
 - d. En la sección Resumen, copia el valor del ARN. Lo necesitará más adelante.

Ahora ha creado el rol de pila con los permisos adecuados y ha obtenido su ARN.

Paso 3: Añada AWS funciones a CodeCatalyst

En este paso, agrega el rol de creación (`codecatalyst-build-role`) y el rol de implementación (`codecatalyst-deploy-role`) a la conexión de CodeCatalyst cuentas de su espacio.

Note

No es necesario añadir el stack role (`codecatalyst-stack-role`) a la conexión. Esto se debe a que el rol de pila lo usa CloudFormation (no CodeCatalyst), después de que ya se haya establecido una conexión entre CodeCatalyst y AWS utilizando el rol de despliegue. Como el rol de pila no lo utiliza CodeCatalyst para acceder a él AWS, no es necesario que esté asociado a una conexión de cuenta.

Para añadir funciones de creación e implementación a la conexión de tu cuenta

1. En CodeCatalyst, navega hasta tu espacio.
2. Elige AWS cuentas. Aparece una lista de conexiones de cuentas.
3. Elija la conexión de cuenta que represente la AWS cuenta en la que creó las funciones de creación e implementación.

4. Seleccione Administrar roles en la consola de AWS administración.

Aparece la página Añadir función de IAM a Amazon CodeCatalyst Space. Puede que tengas que iniciar sesión para acceder a la página.

5. Selecciona Añadir un rol existente que hayas creado en IAM.

Aparece una lista desplegable. La lista muestra todas las funciones de IAM con una política de confianza que incluye a los directores de `codecatalyst.amazonaws.com` servicio `codecatalyst-runner.amazonaws.com` y a los directores de servicio.

6. En la lista desplegable `codecatalyst-build-role`, elija y elija Agregar función.

7. Seleccione Añadir función de IAM, seleccione Añadir una función existente que haya creado en IAM y, en la lista desplegable, elija `codecatalyst-deploy-role` Seleccione Add role (Añadir rol).

Ahora ha agregado las funciones de creación e implementación a su espacio.

8. Copia el valor del nombre CodeCatalyst mostrado de Amazon. Necesitarás este valor más adelante, al crear tu flujo de trabajo.

Paso 4: Crear un bucket de Amazon S3

En este paso, debe crear un depósito de Amazon S3 en el que almacenar el archivo.zip del paquete de despliegue de la aplicación sin servidor.

Para crear un bucket de Amazon S3

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. En el panel principal, selecciona Crear depósito.
3. En el nombre del bucket, introduce:

```
codecatalyst-cfn-s3-bucket
```

4. En AWS Región, seleccione una región. En este tutorial se asume que ha elegido US West (Oregon) `us-west-2`. Para obtener información sobre las regiones compatibles con Amazon S3, consulte los [puntos de enlace y las cuotas de Amazon Simple Storage Service](#) en. Referencia general de AWS
5. En la parte inferior de la página, selecciona Crear depósito.

Ahora ha creado un bucket denominado **codecatalyst-cfn-s3-bucket** en la región us-west-2 de EE. UU. Oeste (Oregón).

Paso 5: Añadir los archivos fuente

En este paso, añadirá varios archivos fuente de la aplicación a su repositorio CodeCatalyst de código fuente. La `hello-world` carpeta contiene los archivos de la aplicación que va a implementar. La `tests` carpeta contiene las pruebas unitarias. La estructura de carpetas es la siguiente:

```
.
|- hello-world
|  |- tests
|    |- unit
|      |- test-handler.js
|  |- app.js
|- .npmignore
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

archivo.npmignore

El `.npmignore` archivo indica qué archivos y carpetas debe excluir npm del paquete de la aplicación. En este tutorial, npm excluye la `tests` carpeta porque no forma parte de la aplicación.

Para agregar el archivo.npmignore

1. [Abra la consola en https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Elige tu proyecto, `codecatalyst-cfn-project`
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. De la lista de repositorios de origen, elige tu repositorio, `codecatalyst-cfn-source-repository`.
5. En Archivos, selecciona Crear archivo.
6. En Nombre de archivo, introduzca:

```
.npmignore
```

7. En el cuadro de texto, introduzca el siguiente código:

```
tests/*
```

8. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora has creado un archivo llamado `.npmignore` en la raíz de tu repositorio.

archivo `package.json`

El `package.json` archivo contiene metadatos importantes sobre tu proyecto de Node, como el nombre del proyecto, el número de versión, la descripción, las dependencias y otros detalles que describen cómo interactuar con la aplicación y cómo ejecutarla.

`package.json` En este tutorial se incluye una lista de dependencias y un `test` script. El script de prueba hace lo siguiente:

- Con [mocha](#), el script de prueba ejecuta las pruebas unitarias especificadas en `hello-world/tests/unit/` y escribe los resultados en un `junit.xml` archivo mediante el reportero [xunit](#).
- En [Estambul \(Nueva York\)](#), el script de prueba genera un informe de cobertura de código (`clover.xml`) utilizando el [Clover](#) Reporter. Para obtener más información, consulte [Uso de reporteros alternativos](#) en la documentación de Estambul.

Para añadir el archivo `package.json`

1. En tu repositorio, en Archivos, selecciona Crear archivo.
2. En Nombre de archivo, introduce:

```
package.json
```

3. En el cuadro de texto, introduzca el siguiente código:

```
{
  "name": "hello_world",
  "version": "1.0.0",
  "description": "hello world sample for NodeJS",
  "main": "app.js",
  "repository": "https://github.com/aws-labs/aws-sam-cli/tree/develop/samcli/local/init/templates/cookiecutter-aws-sam-hello-nodejs",
  "author": "SAM CLI",
  "license": "MIT",
```

```
"dependencies": {
  "axios": "^0.21.1",
  "nyc": "^15.1.0"
},
"scripts": {
  "test": "nyc --reporter=clover mocha hello-world/tests/unit/ --reporter xunit
--reporter-option output=junit.xml"
},
"devDependencies": {
  "aws-sdk": "^2.815.0",
  "chai": "^4.2.0",
  "mocha": "^8.2.1"
}
}
```

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora has añadido un archivo llamado `package.json` a la raíz del repositorio.

archivo `sam-template.yml`

El `sam-template.yml` archivo contiene las instrucciones para implementar la función Lambda y API Gateway y configurarlas juntas. Sigue la [especificación de la AWS Serverless Application Model plantilla](#), que amplía la especificación de la AWS CloudFormation plantilla.

En este tutorial, utilizarás una AWS SAM plantilla en lugar de una AWS CloudFormation plantilla normal porque AWS SAM ofrece un útil tipo de recurso [AWS::Serverless::Function](#). Este tipo realiza una gran parte behind-the-scenes de la configuración que normalmente hay que escribir para usar la sintaxis básica. CloudFormation Por ejemplo, `AWS::Serverless::Function` crea una función Lambda, una función de ejecución de Lambda y asignaciones de fuentes de eventos que inician la función. Tienes que codificar todo esto si quieres escribirlo usando Basic. CloudFormation

Aunque en este tutorial se utiliza una plantilla preescrita, puedes generar una como parte de tu flujo de trabajo mediante una acción de creación. Para obtener más información, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).

Para añadir el archivo `sam-template.yml`

1. En tu repositorio, en Archivos, selecciona Crear archivo.
2. En Nombre de archivo, introduce:

sam-template.yml

3. En el cuadro de texto, introduzca el siguiente código:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  serverless-api

  Sample SAM Template for serverless-api

# More info about Globals: https://github.com/awslabs/serverless-application-model/blob/master/docs/globals.rst
Globals:
  Function:
    Timeout: 3

Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # For details on this resource type,
    see https://github.com/awslabs/serverless-application-model/blob/master/versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: hello-world/
      Handler: app.lambdaHandler
      Runtime: nodejs12.x
      Events:
        HelloWorld:
          Type: Api # For details on this event source type, see
          https://github.com/awslabs/serverless-application-model/blob/master/versions/2016-10-31.md#api
          Properties:
            Path: /hello
            Method: get

Outputs:
  # ServerlessRestApi is an implicit API created out of the events key under
  Serverless::Function
  # Find out about other implicit resources you can reference within AWS SAM at
  # https://github.com/awslabs/serverless-application-model/blob/master/docs/internals/generated_resources.rst#api
  HelloWorldApi:
```

```
Description: "API Gateway endpoint URL for the Hello World function"
Value: !Sub "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/hello/"
HelloWorldFunction:
  Description: "Hello World Lambda function ARN"
  Value: !GetAtt HelloWorldFunction.Arn
HelloWorldFunctionIamRole:
  Description: "Implicit Lambda execution role created for the Hello World
function"
  Value: !GetAtt HelloWorldFunctionRole.Arn
```

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora has añadido un archivo llamado `sam-template.yml` en la carpeta raíz de tu repositorio.

archivo `setup-sam.sh`

El `setup-sam.sh` archivo contiene las instrucciones para descargar e instalar la utilidad AWS SAM CLI. El flujo de trabajo utiliza esta utilidad para empaquetar el `hello-world` código fuente.

Para agregar el archivo `setup-sam.sh`

1. En tu repositorio, en Archivos, selecciona Crear archivo.
2. En Nombre de archivo, introduce:

```
setup-sam.sh
```

3. En el cuadro de texto, introduzca el siguiente código:

```
#!/usr/bin/env bash
echo "Setting up sam"

yum install unzip -y

curl -L0 https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-
linux-x86_64.zip
unzip -qq aws-sam-cli-linux-x86_64.zip -d sam-installation-directory

./sam-installation-directory/install; export AWS_DEFAULT_REGION=us-west-2
```

En el código anterior, sustituye `us-west-2` por tu región. AWS

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora has añadido un archivo llamado `setup-sam.sh` a la raíz del repositorio.

archivo `app.js`

`app.js` Contiene el código de la función Lambda. En este tutorial, el código devuelve el texto `hello world`.

Para añadir el archivo `app.js`

1. En tu repositorio, en Archivos, selecciona Crear archivo.
2. En Nombre de archivo, introduce:

```
hello-world/app.js
```

3. En el cuadro de texto, introduzca el siguiente código:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;

/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 */
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      'statusCode': 200,
```

```
        'body': JSON.stringify({
            message: 'hello world',
            // location: ret.data.trim()
        })
    }
} catch (err) {
    console.log(err);
    return err;
}

return response
};
```

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora ha creado una carpeta llamada `hello-world` y un archivo llamado `app.js`.

archivo `test-handler.js`

El `test-handler.js` archivo contiene pruebas unitarias para la función Lambda.

Para añadir el archivo `test-handler.js`

1. En tu repositorio, en Archivos, selecciona Crear archivo.
2. En Nombre de archivo, introduce:

```
hello-world/tests/unit/test-handler.js
```

3. En el cuadro de texto, introduzca el siguiente código:

```
'use strict';

const app = require('../..//app.js');
const chai = require('chai');
const expect = chai.expect;
var event, context;

describe('Tests index', function () {
    it('verifies successful response', async () => {
        const result = await app.lambdaHandler(event, context)

        expect(result).to.be.an('object');
```

```
    expect(result.statusCode).to.equal(200);
    expect(result.body).to.be.an('string');

    let response = JSON.parse(result.body);

    expect(response).to.be.an('object');
    expect(response.message).to.be.equal("hello world");
    // expect(response.location).to.be.an("string");
  });
});
```

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

Ahora ha agregado un archivo llamado `test-handler.js` en la `hello-world/tests/unit` carpeta.

Ahora ha agregado todos sus archivos fuente.

Tómate un momento para comprobar tu trabajo y asegurarte de que has colocado todos los archivos en las carpetas correctas. La estructura de carpetas es la siguiente:

```
.
|- hello-world
|  |- tests
|    |- unit
|      |- test-handler.js
|  |- app.js
|- .npmignore
|- README.md
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

Paso 6: Crear y ejecutar un flujo de trabajo

En este paso, se crea un flujo de trabajo que empaqueta el código fuente de Lambda y lo despliega. El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).

- Una acción de prueba (Test): al activarse, esta acción instala el [administrador de paquetes Node \(npm\)](#) y, a continuación, ejecuta el `npm run test` comando. Este comando le indica a npm que ejecute el test script definido en el archivo `package.json`. El test script, a su vez, ejecuta las pruebas unitarias y genera dos informes: un informe de prueba (`junit.xml`) y un informe de cobertura de código (`clover.xml`). Para obtener más información, consulte [archivo package.json](#).

A continuación, la acción de prueba transforma los informes XML en CodeCatalyst informes y los muestra en la CodeCatalyst consola, en la pestaña Informes de la acción de prueba.

Para obtener más información sobre la acción de prueba, consulte [Pruebas mediante flujos de trabajo en CodeCatalyst](#).

- Una acción de compilación (BuildBackend): al finalizar la acción de prueba, la acción de compilación descarga e instala la AWS SAM CLI, empaqueta el hello-world código fuente y copia el paquete en su bucket de Amazon S3, donde el servicio Lambda espera que esté. La acción también genera un nuevo archivo de AWS SAM plantilla llamado `sam-template-packaged.yml` y lo coloca en un artefacto de salida llamado `buildArtifact`.

Para obtener más información sobre la acción de creación, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).

- Una acción de despliegue (DeployCloudFormationStack): al finalizar la acción de compilación, la acción de despliegue busca el artefacto de salida generado por la acción de compilación (`buildArtifact`), encuentra la AWS SAM plantilla que contiene y, a continuación, ejecuta la plantilla. La AWS SAM plantilla crea una pila que despliega la aplicación sin servidor.

Para crear un flujo de trabajo

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Seleccione Crear flujo de trabajo.
3. En Repositorio de origen, elija `codecatalyst-cfn-source-repository`.
4. Para Branch, elija `main`.
5. Seleccione Crear.
6. Elimina el código de ejemplo de YAML.
7. Agrega el siguiente código YAML:

```
Name: codecatalyst-cfn-workflow
SchemaVersion: 1.0
```

```
Triggers:
  - Type: PUSH
  Branches:
    - main
Actions:
  Test:
    Identifier: aws/managed-test@v1
    Inputs:
      Sources:
        - WorkflowSource
    Outputs:
      Reports:
        CoverageReport:
          Format: CLOVERXML
          IncludePaths:
            - "coverage/*"
        TestReport:
          Format: JUNITXML
          IncludePaths:
            - junit.xml
    Configuration:
      Steps:
        - Run: npm install
        - Run: npm run test
  BuildBackend:
    Identifier: aws/build@v1
    DependsOn:
      - Test
    Environment:
      Name: codecatalyst-cfn-environment
      Connections:
        - Name: codecatalyst-account-connection
          Role: codecatalyst-build-role
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: . ./setup-sam.sh
        - Run: sam package --template-file sam-template.yml --s3-
bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml
--region us-west-2
    Outputs:
```

```


Artifacts:
  - Name: buildArtifact
    Files:
      - "**/*"
DeployCloudFormationStack:
  Identifier: aws/cfn-deploy@v1
  DependsOn:
    - BuildBackend
  Environment:
    Name: codecatalyst-cfn-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-deploy-role
  Inputs:
    Artifacts:
      - buildArtifact
    Sources: []
  Configuration:
    name: codecatalyst-cfn-stack
    region: us-west-2
    role-arn: arn:aws:iam::111122223333:role/StackRole
    template: ./sam-template-packaged.yml
    capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND

```

En el código anterior, sustituya:

- Ambas instancias *codecatalyst-cfn-environment* con el nombre de su entorno.
- Ambas instancias *codecatalyst-account-connection* con el nombre visible de la conexión de su cuenta. El nombre para mostrar puede ser un número. Para obtener más información, consulte [Paso 3: Añada AWS funciones a CodeCatalyst](#).
- *codecatalyst-build-role* con el nombre del rol de compilación en el que lo creó [Paso 2: Crear AWS roles](#).
- *codecatalyst-cfn-s3 cubos* con el nombre del depósito de Amazon S3 en [Paso 4: Crear un bucket de Amazon S3](#) el que creaste.
- Ambas instancias de *us-west-2* con la región en la que reside su bucket de Amazon S3 (primera instancia) y en la que se implementará su pila (segunda instancia). Estas regiones pueden ser diferentes. En este tutorial se asume que ambas regiones están configuradas en *us-west-2*. Para obtener más información sobre las regiones compatibles con Amazon S3 AWS CloudFormation, consulte los [puntos de enlace y las cuotas del Referencia general de AWSservicio](#) en.

- `codecatalyst-deploy-role` con el nombre de la función de implementación en [Paso 2: Crear AWS roles](#) la que creó.
- `codecatalyst-cfn-environment` con el nombre del entorno en el que ha creado [Requisitos previos](#).
- `arn:aws:iam: :111122223333:role/ con` el nombre de recurso de Amazon (ARN) StackRole del rol de pila en el que creaste. [Paso 2: Crear AWS roles](#)

 Note

Si decidió no crear roles de compilación, implementación y apilamiento `codecatalyst-build-role` `codecatalyst-deploy-role`, sustituya y `arn:aws:iam: :111122223333:role/` por el nombre o el ARN del rol StackRole. CodeCatalystWorkflowDevelopmentRole-`spaceName` Para obtener más información acerca de este rol, consulte [Paso 2: Crear AWS roles](#).

Para obtener información sobre las propiedades del código mostrado anteriormente, consulte la [Referencia de acción «Implementar AWS CloudFormation pila»](#)

8. (Opcional) Selecciona Validar para asegurarte de que el código YAML es válido antes de confirmarlo.
9. Elija Commit (Confirmar).
10. En el cuadro de diálogo Confirmar flujo de trabajo, introduce lo siguiente:
 - a. Para el nombre del archivo de flujo de trabajo, mantenga el valor predeterminado, `codecatalyst-cfn-workflow`.
 - b. Para el mensaje de confirmación, introduzca:

```
add initial workflow file
```

- c. En Repositorio, elija `codecatalyst-cfn-source-repository`.
- d. Para el nombre de la sucursal, elija `principal`.
- e. Elija Commit (Confirmar).

Ahora ha creado un flujo de trabajo. La ejecución de un flujo de trabajo se inicia automáticamente debido al activador definido en la parte superior del flujo de trabajo. En

concreto, cuando depositaste (e insertaste) el `codecatalyst-cfn-workflow.yaml` archivo en tu repositorio de origen, el desencadenador inició la ejecución del flujo de trabajo.

Para ver la ejecución del flujo de trabajo en curso

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el flujo de trabajo que acaba de crear: `codecatalyst-cfn-workflow`
3. Elija la pestaña Ejecuciones.
4. En la columna ID de ejecución, elija el ID de ejecución.
5. Selecciona Probar para ver el progreso de las pruebas.
6. Elige BuildBackendver el progreso de la compilación.
7. Elija DeployCloudFormationStackver el progreso de la implementación.

Para obtener más información sobre la visualización de los detalles de la ejecución, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

8. Cuando finalice la DeployCloudFormationStackacción, haga lo siguiente:
 - Si la ejecución del flujo de trabajo se realizó correctamente, continúe con el siguiente procedimiento.
 - Si la ejecución del flujo de trabajo falló en la prueba o la BuildBackendacción, seleccione Registros para solucionar el problema.
 - Si la ejecución del flujo de trabajo falló en la DeployCloudFormationStackacción, elija la acción de despliegue y, a continuación, elija la pestaña Resumen. Desplázate hasta la sección de CloudFormation eventos para ver el mensaje de error detallado. Si se ha producido una reversión, elimina la `codecatalyst-cfn-stack` pila a través de la AWS CloudFormation consola AWS antes de volver a ejecutar el flujo de trabajo.

Para verificar la implementación

1. Tras una implementación correcta, elija Variables (7) en la barra de menú horizontal situada cerca de la parte superior. (No elija Variables en el panel de la derecha).
2. Junto a HelloWorldApi, pega la `https://` URL en un navegador.

Aparece un mensaje JSON «Hola mundo» de la función Lambda, que indica que el flujo de trabajo implementó y configuró la función Lambda y API Gateway correctamente.

Tip

Puedes hacer que se CodeCatalyst muestre esta URL en el diagrama de flujo de trabajo con unas cuantas configuraciones pequeñas. Para obtener más información, consulte [Mostrar la URL de la aplicación implementada](#).

Para verificar los resultados de las pruebas unitarias y la cobertura del código

1. En el diagrama de flujo de trabajo, elija Probar y, a continuación, elija Informes.
2. Elija TestReportver los resultados de las pruebas unitarias o, en este caso, los detalles de la cobertura del código de los archivos que se están probando, `app.js` y `test-handler.js`. CoverageReport

Para verificar los recursos desplegados

1. Inicie sesión en la consola de API Gateway AWS Management Console y ábrala en <https://console.aws.amazon.com/apigateway/>.
2. Observe la `codecatalyst-cfn-stackAPI` que creó la AWS SAM plantilla. El nombre de la API proviene del `Configuration/name` valor del archivo de definición del flujo de trabajo (`codecatalyst-cfn-workflow.yaml`).
3. Abra la AWS Lambda consola en <https://console.aws.amazon.com/lambda/>.
4. Seleccione Funciones en el panel de navegación.
5. Elija su función Lambda, `codecatalyst-cfn-stack-HelloWorldFunction-string`
6. Puede ver cómo la API Gateway activa la función. Esta integración se configuró automáticamente según el tipo de AWS SAM `AWS::Serverless::Function` recurso.

Paso 7: Realizar un cambio

En este paso, realiza un cambio en el código fuente de Lambda y lo confirma. Esta confirmación inicia una nueva ejecución del flujo de trabajo. Esta ejecución implementa la nueva función Lambda en un esquema azul-verde que usa la configuración de cambio de tráfico predeterminada especificada en la consola Lambda.

Para realizar un cambio en la fuente Lambda

1. En CodeCatalyst, navegue hasta su proyecto.
2. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
3. Elige tu repositorio de origen `codecatalyst-cfn-source-repository`.
4. Cambie el archivo de la aplicación:
 - a. Elija la carpeta `hello-world`.
 - b. Seleccione el archivo `app.js`.
 - c. Elija Editar.
 - d. En la línea 23, cambie `hello world` a **Tutorial complete!**.
 - e. Elija Confirmar y, a continuación, vuelva a elegir Confirmar.

La confirmación hace que se inicie una ejecución de flujo de trabajo. Esta ejecución fallará porque no has actualizado las pruebas unitarias para que reflejen el cambio de nombre.

5. Actualiza las pruebas unitarias:
 - a. Elija `hello-world\tests\unit\test-handler.js`.
 - b. Elija Editar.
 - c. En la línea 19, cambie `hello world` a **Tutorial complete!**.
 - d. Elija Confirmar y, a continuación, vuelva a elegir Confirmar.

La confirmación hace que se inicie otra ejecución del flujo de trabajo. Esta ejecución se realizará correctamente.

6. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
7. Elija `codecatalyst-cfn-workflow`, a continuación, elija Ejecuciones.
8. Elija el ID de ejecución de la última ejecución. Debería estar aún en progreso.
9. Seleccione Probar y verá el progreso `DeployCloudFormationStack` de la ejecución del flujo de trabajo. `BuildBackend`
10. Cuando finalice el flujo de trabajo, elija Variables (7) cerca de la parte superior.
11. Junto a `HelloWorldApi`, pega la `https://` URL en un navegador.

Aparece un `Tutorial complete!` mensaje en el navegador que indica que la nueva aplicación se ha implementado correctamente.

Limpieza

Limpie los archivos y servicios utilizados en este tutorial para evitar que se le cobre por ellos.

Para limpiar en la CodeCatalyst consola

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Eliminarcodecatalyst-cfn-workflow.
3. Eliminarcodecatalyst-cfn-environment.
4. Eliminarcodecatalyst-cfn-source-repository.
5. Eliminarcodecatalyst-cfn-project.

Para limpiar en el AWS Management Console

1. Limpie en CloudFormation, de la siguiente manera:
 - a. Abra la AWS CloudFormation consola en <https://console.aws.amazon.com/cloudformation>.
 - b. Elimine la codecatalyst-cfn-stack.

Al eliminar la pila, se eliminan todos los recursos del tutorial de los servicios de API Gateway y Lambda.

2. Limpie en Amazon S3 de la siguiente manera:
 - a. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
 - b. Elija el icono codecatalyst-cfn-s3-bucket.
 - c. Elimine el contenido del depósito.
 - d. Elimine el bucket.
3. Limpie en IAM de la siguiente manera:
 - a. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - b. Elimine la codecatalyst-deploy-policy.
 - c. Elimine la codecatalyst-build-policy.
 - d. Elimine la codecatalyst-stack-policy.
 - e. Elimine la codecatalyst-deploy-role.
 - f. Elimine la codecatalyst-build-role.

- g. Elimine la `codecatalyst-stack-role`.

En este tutorial, aprendió a implementar una aplicación sin servidor como una CloudFormation pila mediante un CodeCatalyst flujo de trabajo y una acción de implementación de una AWS CloudFormation pila.

Tutorial: Implementación de una aplicación en Amazon ECS

En este tutorial, aprenderá a implementar una aplicación sin servidor en Amazon Elastic Container Service (Amazon ECS) mediante un flujo de trabajo, Amazon ECS y algunos otros servicios. AWS La aplicación implementada es un sencillo sitio web de Hello World creado a partir de una imagen de Docker del servidor web Apache. El tutorial explica el trabajo de preparación necesario, como la configuración de un clúster, y luego describe cómo crear un flujo de trabajo para crear e implementar la aplicación.

Tip

En lugar de seguir este tutorial, puede utilizar un plano que realice una configuración completa de Amazon ECS por usted. Deberá utilizar la API de Node.js con AWS Fargate o la API de Java con el AWS Fargate blueprint. Para obtener más información, consulte [Crear un proyecto con un plano](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configura un AWS usuario y AWS CloudShell](#)
- [Paso 2: implementar una aplicación de marcador de posición en Amazon ECS](#)
- [Paso 3: Crear un repositorio de imágenes de Amazon ECR](#)
- [Paso 4: Crear AWS roles](#)
- [Paso 5: Añadir AWS funciones a CodeCatalyst](#)
- [Paso 6: Crea un repositorio de código fuente](#)
- [Paso 7: Añadir los archivos fuente](#)
- [Paso 8: Crear y ejecutar un flujo de trabajo](#)
- [Paso 9: Realice un cambio en los archivos fuente](#)
- [Limpieza](#)

Requisitos previos

Antes de empezar

- Necesitas un CodeCatalyst espacio con una AWS cuenta conectada. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- En tu espacio, necesitas un CodeCatalyst proyecto vacío para empezar desde cero llamado:

```
codecatalyst-ecs-project
```

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

- En tu proyecto, necesitas un CodeCatalyst entorno llamado:

```
codecatalyst-ecs-environment
```

Configure este entorno de la siguiente manera:

- Elija cualquier tipo, como el de no producción.
- Conecta tu AWS cuenta a ella.

Para obtener más información, consulte [Trabajo con entornos](#).

Paso 1: Configura un AWS usuario y AWS CloudShell

El primer paso de este tutorial consiste en AWS IAM Identity Center crear un usuario y lanzar una AWS CloudShell instancia como este usuario. Durante este tutorial, CloudShell es tu ordenador de desarrollo y es donde configuras AWS los recursos y servicios. Elimine este usuario después de completar el tutorial.

Note

No utilice su usuario root para este tutorial. Debe crear un usuario independiente o, de lo contrario, podría tener problemas al realizar acciones en la AWS Command Line Interface (CLI) más adelante.

Para obtener más información sobre los usuarios del IAM Identity Center CloudShell, consulte la Guía del AWS IAM Identity Center usuario y la Guía AWS CloudShell del usuario.

Para crear un usuario del Centro de Identidad de IAM

1. Inicie sesión en la AWS IAM Identity Center consola AWS Management Console y ábrala en <https://console.aws.amazon.com/singlesignon/>.

Note

Asegúrate de iniciar sesión con el Cuenta de AWS que está conectado a tu CodeCatalyst espacio. Para comprobar qué cuenta está conectada, vaya a su espacio y elija la pestaña Cuentas de AWS. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).

2. En el panel de navegación, elija Users y luego elija la opción Add user.
3. En Nombre de usuario, introduzca:

CodeCatalystECSUser

4. En Contraseña, selecciona Generar una contraseña de un solo uso que puedas compartir con este usuario.
5. En Dirección de correo electrónico y Confirmar dirección de correo electrónico, introduzca una dirección de correo electrónico que no exista aún en el Centro de identidades de IAM.
6. En Nombre y apellidos, introduzca:

CodeCatalystECSUser

7. En Nombre para mostrar, mantenga el nombre generado automáticamente:

CodeCatalystECSUser CodeCatalystECSUser

8. Elija Siguiente.
9. En la página Añadir usuario a grupos, seleccione Siguiente.
10. En la página Revisar y agregar usuario, revise la información y elija Agregar usuario.

Aparece un cuadro de diálogo de contraseña de un solo uso.

11. Seleccione Copiar y, a continuación, pegue la información de inicio de sesión, incluida la URL del portal de AWS acceso y la contraseña de un solo uso.
12. Elija Close.

Para crear un conjunto de permisos

Asignará este conjunto de permisos a CodeCatalystECSUser más adelante.

1. En el panel de navegación, elija Conjuntos de permisos y, a continuación, elija Crear conjunto de permisos.
2. Elija un conjunto de permisos predefinido y, a continuación, seleccione AdministratorAccess. Esta política proporciona permisos completos a todos Servicios de AWS.
3. Elija Siguiente.
4. En Nombre del conjunto de permisos, introduzca:

CodeCatalystECSPermissionSet

5. Elija Siguiente.
6. En la página Revisión, revise la información y, a continuación, elija Crear grupo.

Para asignar el conjunto de permisos a CodeCatalyst ECSUser

1. En el panel de navegación, elija y Cuentas de AWS, a continuación, active la casilla de verificación situada junto a la casilla en la Cuenta de AWS que ha iniciado sesión actualmente.
2. Seleccione Asignar usuarios o grupos.
3. Elija la pestaña Users.
4. Seleccione la casilla situada junto a CodeCatalystECSUser.
5. Elija Siguiente.
6. Seleccione la casilla situada junto a CodeCatalystECSPermissionSet.
7. Elija Siguiente.
8. Revise la información y seleccione Enviar.

Ahora tienes asignada CodeCatalystECSUser y CodeCatalystECSPermissionSet a la tuya Cuenta de AWS, uniéndolas.

Para cerrar sesión y volver a iniciarla como CodeCatalyst ECSUser

1. Antes de cerrar sesión, asegúrese de tener la URL del portal de AWS acceso, así como el nombre de usuario y la contraseña de un solo uso. CodeCatalystECSUser Deberías haber copiado esta información en un editor de texto anteriormente.

Note

Si no dispone de esta información, vaya a la página de CodeCatalystECSUser detalles del Centro de Identidad de IAM, seleccione Restablecer la contraseña, Generar una contraseña de un solo uso [...] y vuelva a restablecer la contraseña para que aparezca la información en la pantalla.

2. Cerrar sesión en AWS.
3. Pegue la URL del portal de AWS acceso en la barra de direcciones del navegador.
4. Inicie sesión con el nombre de usuario y la contraseña de un solo uso deCodeCatalystECSUser.
5. En Nueva contraseña, introduce una contraseña y selecciona Establecer nueva contraseña.

Aparece un cuadro de Cuenta de AWS en la pantalla.

6. Elija y Cuenta de AWS, a continuación, elija el nombre de la persona Cuenta de AWS a la que ha asignado el CodeCatalystECSUser usuario y el conjunto de permisos.
7. Junto a CodeCatalystECSPermissionSet, seleccione Consola de administración.

AWS Management Console Aparece el. Ahora ha iniciado sesión CodeCatalystECSUser con los permisos correspondientes.

Para lanzar una AWS CloudShell instancia

1. Pues CodeCatalystECSUser bien, en la barra de navegación superior, selecciona el AWS icono



).

AWS Management Console Aparece la página principal del.

2. En la barra de navegación superior, selecciona el AWS CloudShell icono



).

CloudShell se abre. Espere a que se cree el CloudShell entorno.

Note

Si no ves el CloudShell icono, asegúrate de que estás en una [región compatible con CloudShell](#). En este tutorial se da por sentado que está en la región de Oeste de EE. UU. (Oregón).

Para comprobar que AWS CLI está instalado

1. En la CloudShell terminal, introduzca:

```
aws --version
```

2. Compruebe que aparezca una versión.

Ya AWS CLI está configurada para el usuario actual `CodeCatalystECSUser`, por lo que no es necesario configurar AWS CLI claves y credenciales, como suele ser el caso.

Paso 2: implementar una aplicación de marcador de posición en Amazon ECS

En esta sección, implementará manualmente una aplicación de marcador de posición en Amazon ECS. Esta aplicación de marcador de posición se sustituirá por la aplicación Hello World implementada en su flujo de trabajo. La aplicación de marcador de posición es Apache Web Server.

Para obtener más información sobre Amazon ECS, consulte la Guía para desarrolladores de Amazon Elastic Container Service.

Complete la siguiente serie de procedimientos para implementar la aplicación de marcador de posición.

Para crear el rol de ejecución de tareas

Esta función otorga a Amazon ECS AWS Fargate (Fargate) permiso para realizar llamadas a la API en su nombre.

1. Cree una política de confianza:
 - a. En AWS CloudShell, introduzca el siguiente comando:

```
cat > codecatalyst-ecs-trust-policy.json
```

Aparece un mensaje parpadeante en la CloudShell terminal.

- b. Introduzca el siguiente código cuando se le solicite:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- c. Coloque el cursor después del último corchete (}).
- d. Pulse **Enter** y, **Ctrl+d** a continuación, para guardar el archivo y salir de cat.

2. Cree un rol de ejecución de tareas:

```
aws iam create-role \
  --role-name codecatalyst-ecs-task-execution-role \
  --assume-role-policy-document file:///codecatalyst-ecs-trust-policy.json
```

3. Adjunte la AmazonECSTaskExecutionRolePolicy política AWS gestionada al rol:

```
aws iam attach-role-policy \
  --role-name codecatalyst-ecs-task-execution-role \
  --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
```

4. Muestra los detalles del rol:

```
aws iam get-role \
  --role-name codecatalyst-ecs-task-execution-role
```

5. Anote el "Arn": valor del rol, por ejemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role`. Necesitará este nombre de recurso de Amazon (ARN) más adelante.

Para crear un clúster de Amazon ECS

Este clúster contendrá la aplicación de marcadores de posición Apache y, más adelante, la aplicación Hello World.

1. Como `CodeCatalystECSUser`, en AWS CloudShell, crea un clúster vacío:

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

2. (Opcional) Compruebe que el clúster se haya creado correctamente:

```
aws ecs list-clusters
```

El ARN del `codecatalyst-ecs-cluster` clúster debe aparecer en la lista, lo que indica que la creación se ha realizado correctamente.

Para crear un archivo de definición de tareas

El archivo de definición de tareas indica que se debe ejecutar la imagen Docker (`httpd:2.4`) [del servidor web Apache 2.4](#), de la que se extrae DockerHub.

1. Como `CodeCatalystECSUser`, en AWS CloudShell, crear un archivo de definición de tareas:

```
cat > taskdef.json
```

2. Pegue el siguiente código cuando se le solicite:

```
{
  "executionRoleArn": "arn:aws:iam::111122223333:role/codecatalyst-ecs-task-
execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": "httpd:2.4",
      "essential": true,
      "portMappings": [
```



```
        {
            "hostPort": 80,
            "protocol": "tcp",
            "containerPort": 80
        }
    ]
},
"requiresCompatibilities": [
    "FARGATE"
],
"cpu": "256",
"family": "codecatalyst-ecs-task-def",
"memory": "512",
"networkMode": "awsvpc"
}
```

En el código anterior, sustituya `arn:aws:iam: :111122223333:role/ -role codecatalyst-ecs-task-execution`

con el ARN de la función de ejecución de tareas que indicó en [Para crear el rol de ejecución de tareas](#)

3. Sitúe el cursor después del último corchete (`})`.
4. Pulse **Enter** y, **Ctrl+d** a continuación, para guardar el archivo y salir de `cat`.

Para registrar el archivo de definición de tareas en Amazon ECS

1. Como `CodeCatalystECSUser`, en AWS CloudShell, registre la definición de la tarea:

```
aws ecs register-task-definition \
    --cli-input-json file://taskdef.json
```

2. (Opcional) Compruebe que se haya registrado la definición de la tarea:

```
aws ecs list-task-definitions
```

La definición de la `codecatalyst-ecs-task-def` tarea debe aparecer en la lista.

Para crear el servicio Amazon ECS

El servicio Amazon ECS ejecuta las tareas (y los contenedores Docker asociados) de la aplicación de marcadores de posición Apache y, posteriormente, de la aplicación Hello World.

1. O bien `CodeCatalystECSUser`, cámbiese a la consola de Amazon Elastic Container Service si aún no lo ha hecho.
2. Elija el clúster que creó anteriormente, `codecatalyst-ecs-cluster`.
3. En la pestaña Servicios, selecciona Crear.
4. En la página Crear, haga lo siguiente:
 - a. Mantenga todos los ajustes predeterminados excepto los que se indican a continuación.
 - b. En Launch type (Tipo de lanzamiento), elija FARGATE.
 - c. En Definición de tarea, en la lista desplegable Familia, elija:

`codecatalyst-ecs-task-def`

- d. En Nombre del servicio, introduzca:

`codecatalyst-ecs-service`

- e. Para las tareas deseadas, introduzca:

`3`

En este tutorial, cada tarea lanza un único contenedor de Docker.

- f. Amplíe la sección Redes.
- g. Para VPC, elija cualquier VPC.
- h. En el caso de las subredes, elija cualquier subred.

Note

Especifique solo una subred. Eso es todo lo que se necesita para este tutorial.

Note

Si no tiene una VPC ni una subred, créelas. Consulte [Crear una VPC](#) y [Crear una subred en la VPC en la Guía del usuario de Amazon VPC](#).

- i. En Grupo de seguridad, elija Crear un nuevo grupo de seguridad y, a continuación, haga lo siguiente:
 - i. En Nombre del grupo de seguridad, introduzca:

`codecatalyst-ecs-security-group`
 - ii. En la descripción del grupo de seguridad, introduzca:

`CodeCatalyst ECS security group`
 - iii. Seleccione Añadir regla. En Tipo, elija HTTP y, en Fuente, elija Anywhere.
 - j. En la parte inferior, selecciona Crear.
 - k. Espera a que se cree el servicio. Puede que tarde unos minutos.
5. Seleccione la pestaña Tareas y, a continuación, pulse el botón de actualización. Compruebe que la columna Último estado de las tres tareas esté configurada como En ejecución.

(Opcional) Para comprobar que la aplicación de marcadores de posición de Apache se está ejecutando

1. En la pestaña Tareas, elija una de las tres tareas.
2. En el campo IP pública, elija la dirección abierta.

Aparece una `It Works!` página. Esto indica que el servicio Amazon ECS inició correctamente una tarea que lanzó un contenedor Docker con la imagen de Apache.

En este punto del tutorial, ha implementado manualmente un clúster, un servicio y una definición de tareas de Amazon ECS, así como una aplicación de marcadores de posición de Apache. Con todos estos elementos listos, ya puede crear un flujo de trabajo que sustituya la aplicación de marcadores de posición de Apache por la aplicación Hello World del tutorial.

Paso 3: Crear un repositorio de imágenes de Amazon ECR

En esta sección, creará un repositorio de imágenes privado en Amazon Elastic Container Registry (Amazon ECR). Este repositorio almacena la imagen de Docker del tutorial, que sustituirá a la imagen de marcador de posición de Apache que implementó anteriormente.

Para obtener más información sobre Amazon ECR, consulte la Guía del usuario de Amazon Elastic Container Registry.

Para crear un repositorio de imágenes en Amazon ECR

1. Como `CodeCatalystECSUser`, en AWS CloudShell, crear un repositorio vacío en Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-ecs-image-repo
```

2. Muestra los detalles del repositorio de Amazon ECR:

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-ecs-image-repo
```

3. Anote el `"repositoryUri"`: valor, por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`.

Lo necesitará más adelante cuando añada el repositorio a su flujo de trabajo.

Paso 4: Crear AWS roles

En esta sección, creará las funciones de AWS IAM que necesitará su CodeCatalyst flujo de trabajo para funcionar. Estas funciones son:

- Función de creación: concede a la acción de CodeCatalyst creación (en el flujo de trabajo) permiso para acceder a su AWS cuenta y escribir en Amazon ECR y Amazon EC2.
- Función de implementación: otorga a la acción CodeCatalyst Implementar en ECS (en el flujo de trabajo) el permiso para acceder a su AWS cuenta, Amazon ECS y algunos otros AWS servicios.

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

Note

Para ahorrar tiempo, puede crear un único rol, denominado `CodeCatalystWorkflowDevelopmentRole-spaceName` rol, en lugar de los dos roles enumerados anteriormente. Para obtener más información, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Tenga en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor. En este tutorial se da por sentado que está creando los dos roles enumerados anteriormente.

Para crear los roles de creación e implementación, puede usar el AWS Management Console o el AWS CLI.

AWS Management Console

Para crear las funciones de creación e implementación, complete la siguiente serie de procedimientos.

Para crear un rol de compilación

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Elija Crear política.
 - e. Seleccione la pestaña JSON.
 - f. Elimina el código existente.
 - g. Pegue el siguiente código:

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

```

        "Action": [
            "ecr:*",
            "ec2:*"
        ],
        "Resource": "*"
    }
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-ecs-build-policy

```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```

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

```

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
```

- e. Elija Siguiente.
- f. En Políticas de permisos `codecatalyst-ecs-build-policy`, busque y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-ecs-build-role

- i. En la descripción del rol, introduzca:

CodeCatalyst ECS build role

- j. Seleccione Crear rol.

Ahora ha creado un rol de compilación con una política de permisos y una política de confianza.

3. Obtenga el ARN del rol de compilación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-ecs-build-role`).
 - c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN. Lo necesitará más adelante.

Para crear un rol de despliegue

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Seleccione Crear política.
 - e. Seleccione la pestaña JSON.
 - f. Elimina el código existente.
 - g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
```



```

    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín de la declaración de política de recursos. A continuación, puede delimitar la política con el nombre del recurso una vez que esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-ecs-deploy-policy
```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

- 2. Cree el rol de implementación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. En Políticas de permisos `codecatalyst-ecs-deploy-policy`, busque y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

```
codecatalyst-ecs-deploy-role
```

- i. En la descripción del rol, introduzca:

```
CodeCatalyst ECS deploy role
```

- j. Seleccione Crear rol.

Ahora ha creado un rol de despliegue con una política de confianza.

3. Obtenga el ARN del rol de implementación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (codecatalyst-ecs-deploy-role).
 - c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN. Lo necesitará más adelante.

AWS CLI

Para crear las funciones de creación e implementación, complete la siguiente serie de procedimientos.

Para crear una política de confianza para ambos roles

ComoCodeCatalystECSUser, en AWS CloudShell, crear un archivo de política de confianza:

1. Cree el archivo:

```
cat > codecatalyst-ecs-trust-policy.json
```

2. En la línea de comandos del terminal, pegue el siguiente código:

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

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

3. Coloque el cursor después del último corchete (}).
4. Pulse **Enter** y, **Ctrl+d** a continuación, para guardar el archivo y salir de cat.

Para crear la política de creación y el rol de creación

1. Cree la política de construcción:
 - a. ComoCodeCatalystECSUser, en AWS CloudShell, crea un archivo de política de compilación:

```
cat > codecatalyst-ecs-build-policy.json
```

- b. Cuando se le solicite, introduzca el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

- c. Coloque el cursor después del último corchete (}).
 - d. Pulse **Enter** y, **Ctrl+d** a continuación, para guardar el archivo y salir de cat.

2. Añada la política de compilación a AWS:

```
aws iam create-policy \  
  --policy-name codecatalyst-ecs-build-policy \  
  --policy-document file://codecatalyst-ecs-build-policy.json
```

3. En el resultado del comando, anote el "arn": valor, por ejemplo, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy`. Necesitará este ARN más adelante.

4. Cree el rol de creación y asígnele la política de confianza:

```
aws iam create-role \  
  --role-name codecatalyst-ecs-build-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

5. Adjunte la política de creación a la función de creación:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy
```

Donde *arn:aws:iam::111122223333:policy/* se sustituye por el ARN de la política de compilación *codecatalyst-ecs-build-policy* que mencionaste anteriormente.

6. Muestra los detalles del rol de compilación:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-build-role
```

7. Anote el "Arn": valor del rol, por ejemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role`. Necesitará este ARN más adelante.

Para crear la política de despliegue y el rol de despliegue

1. Cree una política de despliegue:

a. En AWS CloudShell, cree un archivo de política de despliegue:

```
cat > codecatalyst-ecs-deploy-policy.json
```

- b. Cuando se le solicite, introduzca el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
      "codedeploy:ListApplications",
      "codedeploy:ListDeploymentGroups",
      "codedeploy:ListDeployments",
      "codedeploy:StopDeployment",
      "codedeploy:GetDeploymentTarget",
      "codedeploy:ListDeploymentTargets",
      "codedeploy:GetDeploymentConfig",
      "codedeploy:GetApplicationRevision",
      "codedeploy:RegisterApplicationRevision",
```

```

        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }, { "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": { "StringLike": { "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "codedeploy.amazonaws.com"
    ]
    }
  }
}
]]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- c. Sitúe el cursor tras el último corchete (}).
 - d. Pulse **Enter** y, **Ctrl+d** a continuación, para guardar el archivo y salir de cat.
2. Añada la política de despliegue a AWS:

```

aws iam create-policy \
  --policy-name codecatalyst-ecs-deploy-policy \
  --policy-document file://codecatalyst-ecs-deploy-policy.json

```

3. En el resultado del comando, anote el "arn": valor de la política de despliegue, por ejemplo, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy`. Necesitará este ARN más adelante.
4. Cree el rol de despliegue y asígnele la política de confianza:

```
aws iam create-role \  
  --role-name codecatalyst-ecs-deploy-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

5. Adjunte la política de implementación a la función de implementación, donde `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` se reemplaza por el ARN de la política de implementación que indicó anteriormente.

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-deploy-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy
```

6. Muestra los detalles del rol de despliegue:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-deploy-role
```

7. Anote el "Arn": valor del rol, por ejemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role`. Necesitará este ARN más adelante.

Paso 5: Añadir AWS funciones a CodeCatalyst

En este paso, debe añadir el rol de compilación (`codecatalyst-ecs-build-role`) y el rol de implementación (`codecatalyst-ecs-deploy-role`) a la conexión de CodeCatalyst cuentas de su espacio.

Para añadir funciones de creación e implementación a la conexión de tu cuenta

1. En CodeCatalyst, navega hasta tu espacio.
2. Elige AWS cuentas. Aparece una lista de conexiones de cuentas.
3. Elija la conexión de cuenta que represente la AWS cuenta en la que creó las funciones de creación e implementación.
4. Seleccione Administrar roles en la consola de AWS administración.

Aparece la página Añadir función de IAM a Amazon CodeCatalyst Space. Puede que tengas que iniciar sesión para acceder a la página.

5. Selecciona Añadir un rol existente que hayas creado en IAM.

Aparece una lista desplegable. La lista muestra todas las funciones de IAM con una política de confianza que incluye a los directores de `codecatalyst.amazonaws.com` servicio `codecatalyst-runner.amazonaws.com` y a los directores de servicio.

6. En la lista desplegable `codecatalyst-ecs-build-role`, elija y elija Agregar función.

Note

Si lo ve `The security token included in the request is invalid`, es posible que se deba a que no tiene los permisos adecuados. Para solucionar este problema, cierra sesión AWS como vuelve a iniciar sesión con la AWS cuenta que utilizaste al crear tu CodeCatalyst espacio.

7. Selecciona Añadir función de IAM, selecciona Añadir una función existente que hayas creado en IAM y, en la lista desplegable, selecciona `codecatalyst-ecs-deploy-role` Seleccione Add role (Añadir rol).

Ahora ha agregado las funciones de creación e implementación a su espacio.

8. Copia el valor del nombre CodeCatalyst mostrado de Amazon. Necesitarás este valor más adelante, al crear tu flujo de trabajo.

Paso 6: Crea un repositorio de código fuente

En este paso, crearás un repositorio de fuentes en CodeCatalyst. Este repositorio almacena los archivos fuente del tutorial, como el archivo de definición de tareas.

Para obtener más información sobre los repositorios de fuentes, consulte [Crear un repositorio de código fuente](#).

Para crear un repositorio de origen

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto, `codecatalyst-ecs-project`.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.

4. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
5. En Nombre del repositorio, ingresa:

```
codecatalyst-ecs-source-repository
```

6. Seleccione Crear.

Paso 7: Añadir los archivos fuente

En esta sección, añadirá los archivos fuente de Hello World a su CodeCatalyst repositorio, `codecatalyst-ecs-source-repository`. Se componen de:

- Un `index.html` archivo: muestra un mensaje de Hello World en el navegador.
- Un archivo de Docker: describe la imagen base que se debe usar para la imagen de Docker y los comandos de Docker que se le aplicarán.
- Un `taskdef.json` archivo: define la imagen de Docker que se utilizará al lanzar tareas en el clúster.

La estructura de carpetas es la siguiente:

```
.
|- public-html
|  |- index.html
|- Dockerfile
|- taskdef.json
```

Note

Las siguientes instrucciones le muestran cómo añadir los archivos mediante la CodeCatalyst consola, pero puede utilizar Git si lo prefiere. Para obtener más detalles, consulte [Clonación de un repositorio fuente](#).

Temas

- [index.html](#)
- [Dockerfile](#)

- [taskdef.json](#)


index.html

El `index.html` archivo muestra un mensaje de Hello World en el navegador.

Para añadir el archivo `index.html`

1. En la CodeCatalyst consola, vaya al repositorio de código fuente, `codecatalyst-ecs-source-repository`.
2. En Archivos, selecciona Crear archivo.
3. En Nombre de archivo, introduzca:

```
public-html/index.html
```

 Important

Asegúrese de incluir el `public-html/` prefijo para crear una carpeta con el mismo nombre. `index.html` Se espera que esté en esta carpeta.

4. En el cuadro de texto, introduzca el siguiente código:

```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

5. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

`index.html` Se añade a tu repositorio en una `public-html` carpeta.

Dockerfile

El Dockerfile describe la imagen base de Docker que se debe utilizar y los comandos de Docker que se deben aplicar a ella. [Para obtener más información sobre el Dockerfile, consulta la Referencia de Dockerfile.](#)

El Dockerfile especificado aquí indica que se debe usar la imagen base de Apache 2.4 (`httpd`). También incluye instrucciones para copiar un archivo fuente llamado `index.html` a una carpeta del servidor Apache que sirve páginas web. La `EXPOSE` instrucción del Dockerfile indica a Docker que el contenedor está escuchando en el puerto 80.

Para añadir el Dockerfile

1. En el repositorio de origen, selecciona Crear archivo.
2. En Nombre de archivo, ingresa:

Dockerfile

No incluya una extensión de archivo.

Important

El Dockerfile debe residir en la carpeta raíz del repositorio. El `Docker build` comando del flujo de trabajo espera que esté ahí.

3. En el cuadro de texto, introduzca el siguiente código:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Selecciona Confirmar y, a continuación, selecciona Confirmar de nuevo.

El Dockerfile se añade a tu repositorio.

taskdef.json

El `taskdef.json` archivo que añada en este paso es el mismo que ya especificó, [Paso 2: implementar una aplicación de marcador de posición en Amazon ECS](#) con la siguiente diferencia:

En lugar de especificar un nombre de imagen de Docker codificado en el `image:` campo (`httpd:2.4`), la definición de la tarea aquí utiliza un par de variables para indicar la imagen: `y.$REPOSITORY_URI $IMAGE_TAG`. Estas variables se sustituirán por valores reales generados por la acción de creación del flujo de trabajo cuando ejecute el flujo de trabajo en un paso posterior.

Para obtener más información sobre los parámetros de definición de tareas, consulte [Parámetros de definición de tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Para añadir el archivo `taskdef.json`

1. En tu repositorio de origen, selecciona Crear archivo.
2. En Nombre de archivo, ingresa:

```
taskdef.json
```

3. En el cuadro de texto, introduzca el siguiente código:

```
{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
  execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      # The $REPOSITORY_URI and $IMAGE_TAG variables will be replaced
      # by the workflow at build time (see the build action in the
      # workflow)
      "image": $REPOSITORY_URI:$IMAGE_TAG,
      "essential": true,
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
          "containerPort": 80
        }
      ]
    }
  ]
},
```

```
"requiresCompatibilities": [
  "FARGATE"
],
"networkMode": "awsvpc",
"cpu": "256",
"memory": "512",
"family": "codecatalyst-ecs-task-def"
}
```

En el código anterior, sustituya

arn:aws:iam: :account_id:role/ -role codecatalyst-ecs-task-execution

con el ARN de la función de ejecución de tareas que indicó en. [Para crear el rol de ejecución de tareas](#)

4. Elija Confirmar y, a continuación, vuelva a elegir Confirmar.

El `taskdef.json` archivo se añade a tu repositorio.

Paso 8: Crear y ejecutar un flujo de trabajo

En este paso, crea un flujo de trabajo que toma los archivos fuente, los compila en una imagen de Docker y, a continuación, implementa la imagen en su clúster de Amazon ECS. Esta implementación reemplaza a la aplicación de marcadores de posición de Apache existente.

El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de compilación (BuildBackend): al activarse, la acción crea la imagen de Docker mediante el Dockerfile y envía la imagen a Amazon ECR. La acción de creación también actualiza el `image` campo `taskdef.json` con el valor correcto y, a continuación, crea un artefacto de salida de este archivo. Este artefacto se utiliza como entrada para la acción de despliegue, que es la siguiente.

Para obtener más información sobre la acción de creación, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).

- Una acción de despliegue (DeployToECS): al finalizar la acción de compilación, la acción de despliegue busca el artefacto de salida generado por la acción de compilación (TaskDefArtifact), encuentra su `taskdef.json` interior y lo registra en el servicio Amazon ECS. A continuación, el servicio sigue las instrucciones del `taskdef.json` archivo para ejecutar tres tareas de Amazon ECS (y los contenedores Docker de Hello World asociados) dentro de su clúster de Amazon ECS.

Para crear un flujo de trabajo

1. En la CodeCatalyst consola, en el panel de navegación, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
2. Seleccione Crear flujo de trabajo.
3. En Repositorio de origen, elija `codecatalyst-ecs-source-repository`.
4. Para Branch, elija `main`.
5. Seleccione Crear.
6. Elimina el código de ejemplo de YAML.
7. Agrega el siguiente código YAML:

```
Name: codecatalyst-ecs-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildBackend:
    Identifier: aws/build@v1
    Environment:
      Name: codecatalyst-ecs-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-ecs-build-role
  Inputs:
    Sources:
      - WorkflowSource
    Variables:
      - Name: REPOSITORY_URI
```

```
Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-  
image-repo  
  - Name: IMAGE_TAG  
    Value: ${WorkflowSource.CommitId}  
Configuration:  
Steps:  
  #pre_build:  
  - Run: echo Logging in to Amazon ECR...  
  - Run: aws --version  
  - Run: aws ecr get-login-password --region us-west-2 | docker login --  
username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com  
  #build:  
  - Run: echo Build started on `date`  
  - Run: echo Building the Docker image...  
  - Run: docker build -t $REPOSITORY_URI:latest .  
  - Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG  
  #post_build:  
  - Run: echo Build completed on `date`  
  - Run: echo Pushing the Docker images...  
  - Run: docker push $REPOSITORY_URI:latest  
  - Run: docker push $REPOSITORY_URI:$IMAGE_TAG  
  # Replace the variables in taskdef.json  
  - Run: find taskdef.json -type f | xargs sed -i "s|\$REPOSITORY_URI|  
$REPOSITORY_URI|g"  
  - Run: find taskdef.json -type f | xargs sed -i "s|\$IMAGE_TAG|$IMAGE_TAG|  
g"  
  - Run: cat taskdef.json  
  # The output artifact will be a zip file that contains a task definition  
file.  
Outputs:  
  Artifacts:  
    - Name: TaskDefArtifact  
      Files:  
        - taskdef.json  
DeployToECS:  
  DependsOn:  
    - BuildBackend  
  Identifier: aws/ecs-deploy@v1  
  Environment:  
    Name: codecatalyst-ecs-environment  
  Connections:  
    - Name: codecatalyst-account-connection  
      Role: codecatalyst-ecs-deploy-role  
Inputs:
```



```
Sources: []
Artifacts:
  - TaskDefArtifact
Configuration:
  region: us-west-2
  cluster: codecatalyst-ecs-cluster
  service: codecatalyst-ecs-service
  task-definition: taskdef.json
```

En el código anterior, sustituya:

- Ambas instancias *codecatalyst-ecs-environment* con el nombre del entorno en el que creó [Requisitos previos](#).
- Ambas instancias *codecatalyst-account-connection* con el nombre visible de la conexión de su cuenta. El nombre para mostrar puede ser un número. Para obtener más información, consulte [Paso 5: Añadir AWS funciones a CodeCatalyst](#).
- *codecatalyst-ecs-build-role* con el nombre del rol de compilación en el que creaste [Paso 4: Crear AWS roles](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo* (en la Value: propiedad) con el URI del repositorio de Amazon ECR en el que creó. [Paso 3: Crear un repositorio de imágenes de Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (en el Run: aws ecr comando) con el URI del repositorio Amazon ECR sin el sufijo de imagen (). */codecatalyst-ecs-image-repo*
- *codecatalyst-ecs-deploy-role* con el nombre de la función de implementación en la que creaste. [Paso 4: Crear AWS roles](#)
- Ambas instancias de *us-west-2* con AWS tu código de región. Para obtener una lista de los códigos de región, consulta los [puntos finales regionales en](#). Referencia general de AWS

Note

Si ha decidido no crear funciones de creación e implementación, sustituya *codecatalyst-ecs-build-role* y *codecatalyst-ecs-deploy-role* por el nombre de la CodeCatalystWorkflowDevelopmentRole-*spaceName* función. Para obtener más información acerca de este rol, consulte [Paso 4: Crear AWS roles](#).

Tip

En lugar de utilizar los `sed` comandos `find` y que se muestran en el código del flujo de trabajo anterior para actualizar el repositorio y el nombre de la imagen, puede utilizar la acción de definición de tareas `Render Amazon ECS` para este fin. Para obtener más información, consulte [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#).

8. (Opcional) Elija `Validar` para asegurarse de que el código YAML es válido antes de confirmarlo.
9. Elija `Commit` (Confirmar).
10. En el cuadro de diálogo `Confirmar flujo de trabajo`, introduce lo siguiente:
 - a. En el caso del mensaje de confirmación, elimine el texto e introduzca:

Add first workflow

- b. En `Repositorio`, elija `codecatalyst-ecs-source-repository`.
- c. Para el nombre de la sucursal, elija `principal`.
- d. Elija `Commit` (Confirmar).

Ahora ha creado un flujo de trabajo. La ejecución de un flujo de trabajo se inicia automáticamente debido al activador definido en la parte superior del flujo de trabajo. En concreto, cuando depositaste (e insertaste) el `workflow.yaml` archivo en tu repositorio de origen, el desencadenador inició la ejecución del flujo de trabajo.

Para ver el progreso de la ejecución del flujo de trabajo

1. En el panel de navegación de la CodeCatalyst consola, elija `CI/CD` y, a continuación, elija `Flujos de trabajo`.
2. Elija el flujo de trabajo que acaba de crear, `codecatalyst-ecs-workflow`.
3. Elige `BuildBackend` para ver el progreso de la compilación.
4. Elija `DeployToECS` para ver el progreso de la implementación.

Para obtener más información sobre la visualización de los detalles de la ejecución, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Para verificar el despliegue

1. Abra la consola de Amazon ECS en <https://console.aws.amazon.com/ecs/>.
2. Elija su clúster, `codecatalyst-ecs-cluster`.
3. Elija la pestaña Tareas.
4. Elige una de las tres tareas.
5. En el campo IP pública, elija dirección abierta.

Aparece la página «Hello World» en el navegador, que indica que el servicio Amazon ECS ha implementado correctamente la aplicación.

Paso 9: Realice un cambio en los archivos fuente

En esta sección, realizará un cambio en el `index.html` archivo de su repositorio de origen. Este cambio hace que el flujo de trabajo cree una nueva imagen de Docker, la etiquete con un ID de confirmación, la envíe a Amazon ECR y la despliegue en Amazon ECS.

Para cambiar el archivo `index.html`

1. En la CodeCatalyst consola, en el panel de navegación, selecciona Código, luego elige Repositorios de código fuente y, por último, `codecatalyst-ecs-source-repository` elige tu repositorio.
2. Haga clic en `public.html` y luego en `index.html`.

`index.html` Aparece el contenido de.

3. Elija Editar.
4. En la línea 14, cambie el `Hello World` texto a `Tutorial complete!`.
5. Seleccione Confirmar y, a continuación, vuelva a seleccionar Confirmar.

La confirmación hace que se inicie una nueva ejecución de flujo de trabajo.

6. (Opcional) Ve a la página principal de tu repositorio de origen, selecciona Ver confirmaciones y, a continuación, anota el ID de confirmación del `index.html` cambio.
7. Observe el progreso de la implementación:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija `codecatalyst-ecs-workflow` ver la última ejecución.

- c. Elija BuildBackendy DeployToECS para ver el progreso de la ejecución del flujo de trabajo.
8. Compruebe que la aplicación se haya actualizado de la siguiente manera:
 - a. Abra la consola de Amazon ECS en <https://console.aws.amazon.com/ecs/>.
 - b. Elija su clúster, `codecatalyst-ecs-cluster`.
 - c. Elija la pestaña Tareas.
 - d. Elige una de las tres tareas.
 - e. En el campo IP pública, elija dirección abierta.

Aparece una Tutorial complete! página.
9. (Opcional) En AWS, cambie a la consola Amazon ECR y compruebe que la nueva imagen de Docker esté etiquetada con el ID de confirmación del paso 6.

Limpieza

Limpie los archivos y servicios utilizados en este tutorial para evitar que le cobren por ellos.

En el AWS Management Console, limpie en este orden:

1. En Amazon ECS, haga lo siguiente:
 - a. Eliminar `codecatalyst-ecs-service`.
 - b. Eliminar `codecatalyst-ecs-cluster`.
 - c. Anular registro `codecatalyst-ecs-task-definition`.
2. En Amazon ECR, `codecatalyst-ecs-image-repo` elimínelo.
3. En Amazon EC2, elimínelo. `codecatalyst-ecs-security-group`
4. En el Centro de identidades de IAM, elimine:
 - a. `CodeCatalystECSUser`
 - b. `CodeCatalystECSPermissionSet`

En la CodeCatalyst consola, limpie de la siguiente manera:

1. Eliminar `codecatalyst-ecs-workflow`.
2. Eliminar `codecatalyst-ecs-environment`.
3. Eliminar `codecatalyst-ecs-source-repository`.

4. Eliminarcodecatalyst-ecs-project.

En este tutorial, aprendió a implementar una aplicación en un servicio de Amazon ECS mediante un CodeCatalyst flujo de trabajo y una acción de implementación en Amazon ECS.

Tutorial: Implementación de una aplicación en Amazon EKS

En este tutorial, aprenderá a implementar una aplicación en contenedores en Amazon Elastic Kubernetes Service mediante CodeCatalyst un flujo de trabajo de Amazon, Amazon EKS y algunos otros servicios. AWS La aplicación implementada es un sencillo mensaje de «¡Hola, mundo!» sitio web creado sobre una imagen de Docker del servidor web Apache. El tutorial explica el trabajo de preparación necesario, como la configuración de una máquina de desarrollo y un clúster de Amazon EKS, y luego describe cómo crear un flujo de trabajo para crear la aplicación e implementarla en el clúster.

Una vez finalizada la implementación inicial, el tutorial le indica cómo realizar un cambio en la fuente de la aplicación. Este cambio hace que se cree una nueva imagen de Docker y se envíe a su repositorio de imágenes de Docker con la nueva información de revisión. A continuación, la nueva revisión de la imagen de Docker se implementa en Amazon EKS.

Tip

En lugar de seguir este tutorial, puede utilizar un plano que realice una configuración completa de Amazon EKS por usted. Deberá utilizar el plan de implementación de aplicaciones de EKS. Para obtener más información, consulte [Crear un proyecto con un plano](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configura tu máquina de desarrollo](#)
- [Paso 2: Crear un clúster de Amazon EKS](#)
- [Paso 3: Crear un repositorio de imágenes de Amazon ECR](#)
- [Paso 4: Añadir los archivos fuente](#)
- [Paso 5: Crear AWS roles](#)
- [Paso 6: Añadir AWS roles a CodeCatalyst](#)

- [Paso 7: actualice el ConfigMap](#)
- [Paso 8: Crear y ejecutar un flujo de trabajo](#)
- [Paso 9: Realice un cambio en los archivos fuente](#)
- [Limpieza](#)

Requisitos previos

Antes de comenzar este tutorial:

- Necesitas un CodeCatalyst espacio en Amazon con una AWS cuenta conectada. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- En tu espacio, necesitas un CodeCatalyst proyecto vacío para empezar desde cero llamado:

```
codecatalyst-eks-project
```

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

- En tu proyecto, necesitas un repositorio de CodeCatalyst código fuente vacío llamado:

```
codecatalyst-eks-source-repository
```

Para obtener más información, consulte [Repositorios de fuentes en CodeCatalyst](#).

- En su proyecto, necesita un entorno de CodeCatalyst CI/CD (no un entorno de desarrollo) llamado:

```
codecatalyst-eks-environment
```

Configure este entorno de la siguiente manera:

- Elija cualquier tipo, como el de no producción.
- Conecta tu AWS cuenta a ella.

Para obtener más información, consulte [Trabajo con entornos](#).

Paso 1: Configura tu máquina de desarrollo

El primer paso de este tutorial consiste en configurar una máquina de desarrollo con algunas herramientas que utilizarás a lo largo de este tutorial. Estas herramientas son:

- la `eksctl` utilidad: para la creación de clústeres
- la `kubectl` utilidad: un requisito previo para `eksctl`
- el AWS CLI — también es un requisito previo para `eksctl`

Puede instalar estas herramientas en su máquina de desarrollo actual si tiene una, o puede usar un entorno de CodeCatalyst desarrollo, que está basado en la nube. La ventaja de un CodeCatalyst entorno de desarrollo es que es fácil de configurar y desmontar, y está integrado con otros CodeCatalyst servicios, lo que te permite completar este tutorial en menos pasos.

En este tutorial se asume que utilizarás un entorno CodeCatalyst de desarrollo.

Las siguientes instrucciones describen una forma rápida de lanzar un CodeCatalyst entorno de desarrollo y configurarlo con las herramientas necesarias, pero si desea obtener instrucciones detalladas, consulte:

- [Creación de un entorno de desarrollo](#) en esta guía.
- [Instalación de kubectl en la Guía](#) del usuario de Amazon EKS.
- [Instalación o actualización de eksctl](#) en la Guía del usuario de Amazon EKS.
- [Instalación o actualización de la última versión de la Guía AWS CLI](#) del AWS Command Line Interface usuario.

Para lanzar un entorno de desarrollo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto, `codecatalyst-eks-project`.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. Elige el nombre de tu repositorio fuente, `codecatalyst-eks-source-repository`.
5. En la parte superior, selecciona Crear entorno de desarrollo y, a continuación, selecciona AWS Cloud9 (en el navegador).
6. Asegúrese de que estén seleccionadas la opción Trabajar en la rama existente y la rama principal y, a continuación, elija Crear.

Tu entorno de desarrollo se abre en una nueva pestaña del navegador y tu repositorio (`codecatalyst-eks-source-repository`) se clona en ella.

Para instalar y configurar kubectl

1. En la terminal Dev Environment, introduce:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Introduzca:

```
chmod +x ./kubectl
```

3. Ingresa:

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

4. Ingresa:

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

5. Ingresa:

```
kubectl version --short --client
```

6. Compruebe que aparezca una versión.

Ya lo ha instalado `kubectl`.

Para instalar y configurar eksctl

Note

`eksctl` no es estrictamente obligatorio porque puede usarlo `kubectl` en su lugar. Sin embargo, `eksctl` tiene la ventaja de automatizar gran parte de la configuración del clúster y, por lo tanto, es la herramienta recomendada para este tutorial.

1. En la terminal Dev Environment, introduzca:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```


2. Introduzca:

```
sudo cp /tmp/eksctl /usr/bin
```

3. Ingresa:

```
eksctl version
```

4. Compruebe que aparezca una versión.

Ya lo ha instalado `eksctl`.

Para comprobar que AWS CLI está instalado

1. En la terminal Dev Environment, introduzca:

```
aws --version
```

2. Compruebe que aparezca una versión para comprobar que AWS CLI está instalada.

Complete los procedimientos restantes para configurarlo AWS CLI con los permisos de acceso necesarios AWS.

Para configurar el AWS CLI

Debe configurarlo AWS CLI con claves de acceso y un token de sesión para darle acceso a AWS los servicios. Las siguientes instrucciones proporcionan una forma rápida de configurar las claves y el token, pero si desea obtener instrucciones detalladas, consulte [Configuración de las claves y el token AWS CLI](#) en la Guía del AWS Command Line Interface usuario.

1. Cree un usuario del IAM Identity Center de la siguiente manera:

- a. Inicie sesión en la AWS IAM Identity Center consola AWS Management Console y ábrala en <https://console.aws.amazon.com/singlesignon/>.

(Puede que tengas que seleccionar Activar si nunca has iniciado sesión en el IAM Identity Center).

Note

Asegúrese de iniciar sesión con el Cuenta de AWS que está conectado a su CodeCatalyst espacio. Para comprobar qué cuenta está conectada, vaya a su espacio y elija la pestaña Cuentas de AWS. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).

- b. En el panel de navegación, elija Users y luego elija la opción Add user.
- c. En Nombre de usuario, introduzca:

codecatalyst-eks-user

- d. En Contraseña, selecciona Generar una contraseña de un solo uso que puedas compartir con este usuario.
- e. En Dirección de correo electrónico y Confirmar dirección de correo electrónico, introduzca una dirección de correo electrónico que no exista aún en el Centro de identidades de IAM.
- f. En Nombre, introduzca:

codecatalyst-eks-user

- g. En Apellido, introduzca:

codecatalyst-eks-user

- h. En Nombre para mostrar, mantenga:

codecatalyst-eks-user codecatalyst-eks-user

- i. Elija Siguiente.
- j. En la página Añadir usuario a grupos, selecciona Siguiente.
- k. En la página Revisar y agregar usuario, revise la información y elija Agregar usuario.

Aparece un cuadro de diálogo de contraseña de un solo uso.

- l. Seleccione Copiar y, a continuación, pegue la información de inicio de sesión en un archivo de texto. La información de inicio de sesión consta de la URL del portal de AWS acceso, un nombre de usuario y una contraseña de un solo uso.
- m. Elija Close.

2. Cree un conjunto de permisos, de la siguiente manera:
 - a. En el panel de navegación, elija Conjuntos de permisos y, a continuación, elija Crear conjunto de permisos.
 - b. Elija un conjunto de permisos predefinido y, a continuación, seleccione AdministratorAccess. Esta política proporciona permisos completos a todos Servicios de AWS.
 - c. Elija Siguiente.
 - d. En Nombre del conjunto de permisos, elimine AdministratorAccess e introduzca:

```
codecatalyst-eks-permission-set
```

- e. Elija Siguiente.
 - f. En la página Revisión, revise la información y, a continuación, elija Crear grupo.
3. Asigne el conjunto de permisos a `codecatalyst-eks-user`, de la siguiente manera:

- a. En el panel de navegación, elija y Cuentas de AWS, a continuación, active la casilla de verificación situada junto a la casilla en la Cuenta de AWS que ha iniciado sesión actualmente.
- b. Seleccione Asignar usuarios o grupos.
- c. Elija la pestaña Users.
- d. Seleccione la casilla situada junto a `codecatalyst-eks-user`.
- e. Elija Siguiente.
- f. Seleccione la casilla situada junto a `codecatalyst-eks-permission-set`.
- g. Elija Siguiente.
- h. Revise la información y seleccione Enviar.

Ahora tienes asignada `codecatalyst-eks-user` y `codecatalyst-eks-permission-set` a la tuya Cuenta de AWS, uniéndolas.

4. Obtenga las claves `codecatalyst-eks-user` de acceso y el token de sesión de la siguiente manera:
 - a. Asegúrese de tener la URL del portal de AWS acceso y el nombre de usuario y la contraseña de un solo uso del mismo `codecatalyst-eks-user`. Deberías haber copiado esta información en un editor de texto anteriormente.

Note

Si no dispone de esta información, vaya a la página de `codecatalyst-eks-user` detalles del Centro de Identidad de IAM, seleccione Restablecer la contraseña, Generar una contraseña de un solo uso [...] y vuelva a restablecer la contraseña para que aparezca la información en la pantalla.

- b. Cerrar sesión en AWS.
- c. Pegue la URL del portal de AWS acceso en la barra de direcciones del navegador.
- d. Inicie sesión con:

- Nombre de usuario:

```
codecatalyst-eks-user
```

- Contraseña:

one-time-password

- e. En Establecer nueva contraseña, introduzca una nueva contraseña y elija Establecer nueva contraseña.

Aparece un cuadro de Cuenta de AWS en la pantalla.

- f. Elija y Cuenta de AWS, a continuación, elija el nombre de la persona Cuenta de AWS a la que ha asignado el `codecatalyst-eks-user` usuario y el conjunto de permisos.
- g. Junto a `codecatalyst-eks-permission-set`, seleccione Línea de comandos o Acceso mediante programación.
- h. Copia los comandos en el centro de la página. Tienen un aspecto similar al siguiente:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... donde *session-token* es una cadena aleatoria larga.

5. Añada las claves de acceso y el token de sesión a AWS CLI, de la siguiente manera:
 - a. Regrese a su CodeCatalyst entorno de desarrollo.

- b. En la línea de comandos del terminal, pegue los comandos que copió. Pulse Intro.

Ahora lo ha configurado AWS CLI con claves de acceso y un token de sesión. Ahora puede utilizarla AWS CLI para completar las tareas requeridas en este tutorial.

Important

Si en algún momento de este tutorial ves mensajes similares a:

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```

O bien:

```
ExpiredToken: The security token included in the request is expired
```

... es porque tu AWS CLI sesión ha caducado. En este caso, no ejecute el `aws configure` comando. En su lugar, siga las instrucciones del paso 4 de este procedimiento, que comienza con `Obtain codecatalyst-eks-user's access key and session token` la actualización de la sesión.

Paso 2: Crear un clúster de Amazon EKS

En esta sección, creará un clúster en Amazon EKS. Las instrucciones que aparecen a continuación describen una forma rápida de crear el clúster mediante `eksctl`, pero si desea obtener instrucciones detalladas, consulte:

- [Introducción a eksctl](#) en la Guía del usuario de Amazon EKS
- o
- [Introducción a la consola y](#) a la Guía del usuario de Amazon EKS (AWS CLI en este tema se proporcionan `kubectl` instrucciones para crear el clúster)

Note

La CodeCatalyst integración con Amazon EKS no admite [clústeres privados](#).


Antes de empezar

Asegúrese de haber realizado las siguientes tareas en su máquina de desarrollo:

- Instaló la `eksctl` utilidad.
- Se instaló la `kubectl` utilidad.
- La instalé AWS CLI y la configuré con claves de acceso y un token de sesión.

Para obtener información sobre cómo realizar estas tareas, consulte [Paso 1: Configura tu máquina de desarrollo](#).

Pasos para crear un clúster

 Important

No utilice la interfaz de usuario del servicio Amazon EKS para crear el clúster, ya que el clúster no se configurará correctamente. Utilice la `eksctl` utilidad, tal y como se describe en los pasos siguientes.

1. Ve a tu entorno de desarrollo.
2. Cree un clúster y nodos:


```
eksctl create cluster --name codecatalyst-eks-cluster --region us-west-2
```

Donde:

- *codecatalyst-eks-cluster* se sustituye por el nombre que desee darle al clúster.
- *us-west-2* se sustituye por tu región.

Transcurridos entre 10 y 20 minutos, aparecerá un mensaje similar al siguiente:

```
EKS cluster "codecatalyst-eks-cluster" in "us-west-2" region is ready
```

 Note

Verás varios `waiting for CloudFormation stack` mensajes mientras creas AWS el clúster. Esto es normal.

3. Comprueba que el clúster se haya creado correctamente:

```
kubectl cluster-info
```

Verás un mensaje similar al siguiente, que indica que la creación del clúster se ha realizado correctamente:

```
Kubernetes master is running at https://long-string.gr7.us-west-2.eks.amazonaws.com
CoreDNS is running at https://long-string.gr7.us-west-2.eks.amazonaws.com/api/v1/
namespaces/kube-system/services/kube-dns:dns/proxy
```

Paso 3: Crear un repositorio de imágenes de Amazon ECR

En esta sección, creará un repositorio de imágenes privado en Amazon Elastic Container Registry (Amazon ECR). Este repositorio almacena la imagen de Docker para el tutorial.

Para obtener más información sobre Amazon ECR, consulte la Guía del usuario de Amazon Elastic Container Registry.

Para crear un repositorio de imágenes en Amazon ECR

1. Vaya a su entorno de desarrollo.
2. Cree un repositorio vacío en Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-eks-image-repo
```

codecatalyst-eks-image-repo Sustitúyalo por el nombre que desee asignar al repositorio de Amazon ECR.

En este tutorial se asume que ha asignado un nombre a su repositorio `codecatalyst-eks-image-repo`.

3. Muestra los detalles del repositorio de Amazon ECR:

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-eks-image-repo
```

4. Anote el “`repositoryUri`”: valor, por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo`.

Lo necesitará más adelante cuando añada el repositorio a su flujo de trabajo.

Paso 4: Añadir los archivos fuente

En esta sección, añadirá los archivos fuente de la aplicación a su repositorio de código fuente (codecatalyst-eks-source-repository). Se componen de:

- Un `index.html` archivo: muestra un mensaje de «¡Hola, mundo!» mensaje en el navegador.
- Un archivo de Docker: describe la imagen base que se va a utilizar para la imagen de Docker y los comandos de Docker que se van a aplicar a ella.
- Un `deployment.yaml` archivo: el manifiesto de Kubernetes que define el servicio y la implementación de Kubernetes.

La estructura de carpetas es la siguiente:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
    | |– index.html
  |– Dockerfile
```

Temas

- [index.html](#)
- [Dockerfile](#)
- [deployment.yaml](#)

index.html

El `index.html` archivo muestra el mensaje «¡Hola, mundo!» mensaje en el navegador.

Para añadir el archivo `index.html`

1. Ve a tu entorno de desarrollo.
2. En `codecatalyst-eks-source-repository`, crea una carpeta llamada `public-html`.
3. En `public-html`, cree un archivo llamado `index.html` con el siguiente contenido:


```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

4. En la línea de comandos del terminal, escriba:

```
cd /projects/codecatalyst-eks-source-repository
```

5. Agrega, confirma y presiona:

```
git add .
git commit -m "add public-html/index.html"
git push
```

`index.html` Se añade a tu repositorio en una `public-html` carpeta.

Dockerfile

El Dockerfile describe la imagen base de Docker que se debe utilizar y los comandos de Docker que se deben aplicar a ella. [Para obtener más información sobre el Dockerfile, consulta la Referencia de Dockerfile.](#)

El Dockerfile especificado aquí indica que se debe usar la imagen base de Apache 2.4 (`httpd`). También incluye instrucciones para copiar un archivo fuente llamado `index.html` a una carpeta del servidor Apache que sirve páginas web. La `EXPOSE` instrucción del Dockerfile indica a Docker que el contenedor está escuchando en el puerto 80.

Para añadir el Dockerfile

1. En `encodecatalyst-eks-source-repository`, cree un archivo llamado `Dockerfile` con el siguiente contenido:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

No incluya una extensión de archivo.

Important

El `Dockerfile` debe residir en la carpeta raíz del repositorio. El `Docker build` comando del flujo de trabajo espera que esté ahí.

2. Agrega, confirma y presiona:

```
git add .
git commit -m "add Dockerfile"
git push
```

El `Dockerfile` se añade a tu repositorio.

deployment.yaml

En esta sección, agregas un `deployment.yaml` archivo a tu repositorio. El `deployment.yaml` archivo es un manifiesto de Kubernetes que define dos tipos o tipos de recursos de Kubernetes que se van a ejecutar: un «servicio» y un «despliegue».

- El «servicio» implementa un balanceador de carga en Amazon EC2. El balanceador de carga le proporciona una URL pública con acceso a Internet y un puerto estándar (puerto 80) que puede usar para buscar el mensaje «¡Hola, mundo!» "Hello, World!".
- La «implementación» despliega tres módulos y cada uno de ellos contendrá un contenedor Docker con el mensaje «¡Hola, mundo!» "Hello, World!". Los tres pods se implementan en los nodos que se crearon al crear el clúster.

El manifiesto de este tutorial es breve; sin embargo, un manifiesto puede incluir cualquier tipo de recursos de Kubernetes, como pods, trabajos, entradas y políticas de red. Además, puedes usar varios archivos de manifiesto si tu implementación es compleja.

Para añadir un archivo `deployment.yaml`

1. En `encodecatalyst-eks-source-repository`, crea una carpeta llamada `Kubernetes`
2. En `Kubernetes`, cree un archivo llamado `deployment.yaml` con el siguiente contenido:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-app
spec:
  type: LoadBalancer
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - name: codecatalyst-eks-container
```

```
# The $REPOSITORY_URI and $IMAGE_TAG placeholders will be replaced by
actual values supplied by the build action in your workflow
image: $REPOSITORY_URI:$IMAGE_TAG
ports:
- containerPort: 80
```

3. Agrega, confirma y envía:

```
git add .
git commit -m "add Kubernetes/deployment.yaml"
git push
```

El `deployment.yaml` archivo se añade a tu repositorio en una carpeta llamada `Kubernetes`.

Ahora ha agregado todos sus archivos fuente.

Tómate un momento para comprobar tu trabajo y asegurarte de que has colocado todos los archivos en las carpetas correctas. La estructura de carpetas es la siguiente:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
    | |– index.html
  |– Dockerfile
```

Paso 5: Crear AWS roles

En esta sección, creará las funciones de AWS IAM que necesitará su CodeCatalyst flujo de trabajo para funcionar. Estas funciones son:

- **Función de creación:** concede a la acción de CodeCatalyst creación (en el flujo de trabajo) permiso para acceder a su AWS cuenta y escribir en Amazon ECR y Amazon EC2.
- **Función de implementación:** otorga a la acción CodeCatalyst Implementar en el clúster de Kubernetes (en el flujo de trabajo) el permiso para acceder a su cuenta AWS y a Amazon EKS.

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

Note

Para ahorrar tiempo, puede crear un único rol, denominado `CodeCatalystWorkflowDevelopmentRole-spaceName` rol, en lugar de los dos roles enumerados anteriormente. Para obtener más información, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Tenga en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor. En este tutorial se da por sentado que está creando los dos roles enumerados anteriormente.

Para crear las funciones de creación e implementación, complete la siguiente serie de procedimientos.

1. Para crear una política de confianza para ambos roles

1. Ve a tu entorno de desarrollo.
2. En el Cloud9-*long-string* directorio, cree un archivo llamado `codecatalyst-eks-trust-policy.json` con el siguiente contenido:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Para crear la política de compilación para el rol de compilación

- En el Cloud9-*long-string* directorio, cree un archivo llamado `codecatalyst-eks-build-policy.json` con el siguiente contenido:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

3. Para crear la política de despliegue para el rol de despliegue

- En el Cloud9-*long-string* directorio, cree un archivo denominado `codecatalyst-eks-deploy-policy.json` con el siguiente contenido:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",

```

```

        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

Ahora ha agregado tres documentos de políticas a su entorno de desarrollo. La estructura de su directorio ahora tiene el siguiente aspecto:

```

|- Cloud9-long-string
  |- .c9
  |- codecatalyst-eks-source-repository
    |- Kubernetes
    |- public-html
    |- Dockerfile
  codecatalyst-eks-build-policy.json
  codecatalyst-eks-deploy-policy.json
  codecatalyst-eks-trust-policy.json

```

4. Para añadir la política de compilación a AWS

1. En la terminal Dev Environment, ingresa:

```
cd /projects
```

2. Introduzca:

```
aws iam create-policy \
  --policy-name codecatalyst-eks-build-policy \

```

```
--policy-document file://codecatalyst-eks-build-policy.json
```

3. Pulse Intro.
4. En el resultado del comando, anote el "arn": valor, por ejemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Necesitará este ARN más adelante.

5. Para agregar la política de despliegue a AWS

1. Introduzca:

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-deploy-policy \  
  --policy-document file://codecatalyst-eks-deploy-policy.json
```

2. Pulse Intro.
3. En el resultado del comando, anote el "arn": valor de la política de despliegue, por ejemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy`. Necesitará este ARN más adelante.

6. Para crear el rol de compilación

1. Introduzca:

```
aws iam create-role \  
  --role-name codecatalyst-eks-build-role \  
  --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Pulse Intro.
3. Ingresa:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-eks-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

Donde *arn:aws:iam::111122223333:policy/* se sustituye por el ARN de la política de compilación *codecatalyst-eks-build-policy* que mencionaste anteriormente.

4. Pulse Intro.
5. En la línea de comandos del terminal, ingresa:

```
aws iam get-role \  
    --role-name codecatalyst-eks-build-role
```

6. Pulse Intro.
7. Anote el "Arn": valor del rol, por ejemplo, `arn:aws:iam::111122223333:role/codecatalyst-eks-build-role`. Necesitará este ARN más adelante.

7. Para crear el rol de despliegue

1. Introduzca:

```
aws iam create-role \  
    --role-name codecatalyst-eks-deploy-role \  
    --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Pulse Intro.
3. Ingresa:

```
aws iam attach-role-policy \  
    --role-name codecatalyst-eks-deploy-role \  
    --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy
```

Donde *arn:aws:iam::111122223333:policy/* se sustituye por el ARN de la política de `codecatalyst-eks-deploy-policy` despliegue que indicó anteriormente.

4. Pulse Intro.
5. Introduzca:

```
aws iam get-role \  
    --role-name codecatalyst-eks-deploy-role
```

6. Pulse Intro.
7. Anote el "Arn": valor del rol, por ejemplo, `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Necesitará este ARN más adelante.

Ahora ha creado las funciones de creación e implementación y ha anotado sus ARN.

Paso 6: Añadir AWS roles a CodeCatalyst

En este paso, agrega el rol de compilación (`codecatalyst-eks-build-role`) y el rol de implementación (`codecatalyst-eks-deploy-role`) al rol Cuenta de AWS que conectó a su espacio. Esto hace que los roles estén disponibles para su uso en su flujo de trabajo.

Para añadir funciones de creación e implementación a su Cuenta de AWS

1. En la CodeCatalyst consola, navega hasta tu espacio.
2. En la parte superior, selecciona Configuración.
3. En el panel de navegación, selecciona AWS cuentas. Aparece una lista de cuentas.
4. En la columna Nombre CodeCatalyst para mostrar de Amazon, copia el nombre para mostrar del Cuenta de AWS lugar en el que creaste las funciones de creación e implementación. (Puede ser un número). Necesitarás este valor más adelante, al crear tu flujo de trabajo.
5. Elige el nombre para mostrar.
6. Selecciona Administrar funciones en la consola de AWS administración.

Aparece la página Añadir función de IAM a Amazon CodeCatalyst Space. Puede que tengas que iniciar sesión para acceder a la página.

7. Selecciona Añadir un rol existente que hayas creado en IAM.

Aparece una lista desplegable. La lista muestra las funciones de creación e implementación, así como cualquier otra función de IAM con una política de confianza que incluya las funciones principales `codecatalyst-runner.amazonaws.com` y de `codecatalyst.amazonaws.com` servicio.

8. En la lista desplegable, añade:
 - `codecatalyst-eks-build-role`
 - `codecatalyst-eks-deploy-role`

Note

Si lo ve `The security token included in the request is invalid`, es posible que se deba a que no tiene los permisos adecuados. Para solucionar este

problema, cierra sesión AWS como vuelve a iniciar sesión con la AWS cuenta que utilizaste al crear tu CodeCatalyst espacio.

- Vuelve a la CodeCatalyst consola y actualiza la página.

Las funciones de creación e implementación deberían aparecer ahora en las funciones de IAM.

Estas funciones ya están disponibles para su uso en los CodeCatalyst flujos de trabajo.

Paso 7: actualice el ConfigMap

Debe añadir la función de despliegue que creó al ConfigMap archivo de Kubernetes [Paso 5: Crear AWS roles](#) para que la acción Implementar en el clúster de Kubernetes (en su flujo de trabajo) pueda acceder a su clúster e interactuar con él. Puedes usar `eksctl` o `kubectl` para realizar esta tarea.

Para configurar el archivo de Kubernetes ConfigMap mediante `eksctl`

- En la terminal Dev Environment, introduzca:

```
eksctl create iamidentitymapping --cluster codecatalyst-eks-cluster --
arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role --group
system:masters --username codecatalyst-eks-deploy-role --region us-west-2
```

Donde:

- codecatalyst-eks-cluster* se sustituye por el nombre del clúster de Amazon EKS.
- arn:aws:iam::111122223333:role/* se sustituye por el ARN del rol de despliegue en *codecatalyst-eks-deploy-role* el que creaste. [Paso 5: Crear AWS roles](#)
- codecatalyst-eks-deploy-role* (junto a `--username`) se sustituye por el nombre de la función de despliegue en la que creaste. [Paso 5: Crear AWS roles](#)

Note

Si ha decidido no crear un rol de despliegue, *codecatalyst-eks-deploy-role* sustitúyalo por el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol. Para obtener más información acerca de este rol, consulte [Paso 5: Crear AWS roles](#).

- us-west-2* se sustituye por tu región.

Para obtener más información sobre este comando, consulte [Administrar usuarios y roles de IAM](#).

Aparece un mensaje similar al siguiente:

```
2023-06-09 00:58:29 [#] checking arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role against entries in the auth ConfigMap
2023-06-09 00:58:29 [#] adding identity "arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role" to auth ConfigMap
```

Para configurar el archivo de Kubernetes ConfigMap mediante kubectl

1. En la terminal Dev Environment, ingresa:

```
kubectl edit configmap -n kube-system aws-auth
```

El ConfigMap archivo aparece en la pantalla.

2. Añada el texto en cursiva roja:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file
# will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::111122223333:role/eksctl-codecatalyst-eks-cluster-n-NodeInstanceRole-16BC456ME6YR5
      username: system:node:{{EC2PrivateDNSName}}
    - groups:
      - system:masters
      rolearn: arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role
      username: codecatalyst-eks-deploy-role
  mapUsers: |
    []
```

```
kind: ConfigMap
metadata:
  creationTimestamp: "2023-06-08T19:04:39Z"
  managedFields:
  ...
```

Donde:

- *arn:aws:iam: :111122223333:role/* se sustituye por el ARN del rol de despliegue en `codecatalyst-eks-deploy-role` el que creaste. [Paso 5: Crear AWS roles](#)
- *codecatalyst-eks-deploy-role* (junto a `username:`) se sustituye por el nombre de la función de despliegue en la que creaste. [Paso 5: Crear AWS roles](#)

Note

Si ha decidido no crear un rol de despliegue, *codecatalyst-eks-deploy-role* sustitúyalo por el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol. Para obtener más información acerca de este rol, consulte [Paso 5: Crear AWS roles](#).

Para obtener más información, consulte [Habilitar el acceso principal de IAM a su clúster](#) en la Guía del usuario de Amazon EKS.

Ahora ha asignado la función de implementación y, por extensión, la acción Implementar en Amazon EKS, `system:masters` permisos para su clúster de Kubernetes.

Paso 8: Crear y ejecutar un flujo de trabajo

En este paso, debe crear un flujo de trabajo que toma los archivos fuente, los compila en una imagen de Docker y, a continuación, implementa la imagen en grupos de árboles del clúster de Amazon EKS.

El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).


- Una acción de compilación (BuildBackend): al activarse, la acción crea la imagen de Docker mediante el Dockerfile y envía la imagen a Amazon ECR. La acción de compilación también actualiza las `$IMAGE_TAG` variables `$REPOSITORY_URI` y del `deployment.yaml` archivo con los valores correctos y, a continuación, crea un artefacto de salida de este archivo y de cualquier otro archivo de la carpeta. Kubernetes En este tutorial, el único archivo de la Kubernetes carpeta es `deployment.yaml`, pero puede incluir más archivos. El artefacto se utiliza como entrada para la acción de despliegue, que es la siguiente.

Para obtener más información sobre la acción de construcción, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).

- Una acción de despliegue (DeployToEKS): al finalizar la acción de compilación, la acción de despliegue busca el artefacto de salida generado por la acción de compilación (Manifests) y encuentra el `deployment.yaml` archivo que contiene. A continuación, la acción sigue las instrucciones del `deployment.yaml` archivo para ejecutar tres módulos, cada uno de los cuales contiene un único mensaje: «¡Hola, mundo!» Contenedor Docker: dentro de su clúster de Amazon EKS.

Para crear un flujo de trabajo

1. Vaya a la consola. CodeCatalyst
2. Navega hasta tu proyecto (`codecatalyst-eks-project`).
3. En el panel de navegación, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
4. Seleccione Crear flujo de trabajo.
5. En Repositorio de origen, elija `codecatalyst-eks-source-repository`.
6. Para Branch, elija `main`.
7. Seleccione Crear.
8. Elimina el código de ejemplo de YAML.
9. Agrega el siguiente código YAML para crear un nuevo archivo de definición de flujo de trabajo:

 Note

Para obtener más información sobre el archivo de definición del flujo de trabajo, consulte [Referencia de definición de flujo de trabajo](#).

Name: codecatalyst-eks-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH

Branches:

- main

Actions:

BuildBackend:

Identifier: aws/build@v1

Environment:

Name: *codecatalyst-eks-environment*

Connections:

- Name: *codecatalyst-account-connection*

- Role: *codecatalyst-eks-build-role*

Inputs:

Sources:

- WorkflowSource

Variables:

- Name: REPOSITORY_URI

- Value: *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo*

- Name: IMAGE_TAG

- Value: `${WorkflowSource.CommitId}`

Configuration:

Steps:

#pre_build:

- Run: echo Logging in to Amazon ECR...

- Run: aws --version

- Run: aws ecr get-login-password --region *us-west-2* | docker login --username AWS --password-stdin *111122223333.dkr.ecr.us-west-2.amazonaws.com*

#build:

- Run: echo Build started on `date`

- Run: echo Building the Docker image...

- Run: docker build -t \$REPOSITORY_URI:latest .

- Run: docker tag \$REPOSITORY_URI:latest \$REPOSITORY_URI:\$IMAGE_TAG

#post_build:

- Run: echo Build completed on `date`

- Run: echo Pushing the Docker images...

- Run: docker push \$REPOSITORY_URI:latest

- Run: docker push \$REPOSITORY_URI:\$IMAGE_TAG

Replace the variables in deployment.yaml

```


- Run: find Kubernetes/ -type f | xargs sed -i "s|\\$REPOSITORY_URI|
$REPOSITORY_URI|g"
- Run: find Kubernetes/ -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|g"
- Run: cat Kubernetes/*
# The output artifact will be a zip file that contains Kubernetes manifest
files.
Outputs:
  Artifacts:
    - Name: Manifests
      Files:
        - "Kubernetes/*"
DeployToEKS:
DependsOn:
  - BuildBackend
Identifier: aws/kubernetes-deploy@v1
Environment:
  Name: codecatalyst-eks-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-eks-deploy-role
Inputs:
  Artifacts:
    - Manifests
Configuration:
  Namespace: default
  Region: us-west-2
  Cluster: codecatalyst-eks-cluster
  Manifests: Kubernetes/

```

En el código anterior, sustituya:

- Ambas instancias *codecatalyst-eks-environment* con el nombre del entorno en el que creó [Requisitos previos](#).
- Ambas instancias *codecatalyst-account-connection* con el nombre visible de la conexión de su cuenta. El nombre para mostrar puede ser un número. Para obtener más información, consulte [Paso 6: Añadir AWS roles a CodeCatalyst](#).
- *codecatalyst-eks-build-role* con el nombre del rol de compilación en el que creaste [Paso 5: Crear AWS roles](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo* (en la Value: propiedad) con el URI del repositorio de Amazon ECR en el que creó. [Paso 3: Crear un repositorio de imágenes de Amazon ECR](#)

- `111122223333.dkr.ecr.us-west-2.amazonaws.com` (en el Run: `aws ecr` comando) con el URI del repositorio Amazon ECR sin el sufijo de imagen (`()`). `/codecatalyst-eks-image-repo`
- `codecatalyst-eks-deploy-role` con el nombre de la función de implementación en la que creaste. [Paso 5: Crear AWS roles](#)
- Ambas instancias de `us-west-2` con AWS tu código de región. Para obtener una lista de los códigos de región, consulta los [puntos finales regionales en](#). Referencia general de AWS

 Note

Si ha decidido no crear funciones de creación e implementación, sustituya `codecatalyst-eks-build-role` y `codecatalyst-eks-deploy-role` por el nombre de la `CodeCatalystWorkflowDevelopmentRole-spaceName` función. Para obtener más información acerca de este rol, consulte [Paso 5: Crear AWS roles](#).

10. (Opcional) Selecciona Validar para asegurarte de que el código YAML es válido antes de realizar la confirmación.
11. Elija Commit (Confirmar).
12. En el cuadro de diálogo Confirmar flujo de trabajo, introduce lo siguiente:
 - a. En el caso del mensaje de confirmación, elimine el texto e introduzca:

Add first workflow

- b. En Repositorio, elija `codecatalyst-eks-source-repository`.
- c. En Nombre de sucursal, elija principal.
- d. Elija Commit (Confirmar).

Ahora ha creado un flujo de trabajo. La ejecución de un flujo de trabajo se inicia automáticamente debido al activador definido en la parte superior del flujo de trabajo. En concreto, cuando depositaste (e insertaste) el `workflow.yaml` archivo en tu repositorio de origen, el desencadenador inició la ejecución del flujo de trabajo.

Para ver el progreso de la ejecución del flujo de trabajo

1. En el panel de navegación de la CodeCatalyst consola, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el flujo de trabajo que acaba de crear, `codecatalyst-eks-workflow`
3. Elige BuildBackendver el progreso de la compilación.
4. Elija DeployToEKS para ver el progreso de la implementación.

Para obtener más información sobre la visualización de los detalles de la ejecución, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Para verificar el despliegue

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. A la izquierda, cerca de la parte inferior, selecciona Load Balancers.
3. Selecciona el balanceador de carga que se creó como parte de tu implementación de Kubernetes. Si no estás seguro de qué balanceador de cargas elegir, busca las siguientes etiquetas en la pestaña Etiquetas:
 - `kubernetes.io/service-name`
 - `kubernetes.io/cluster/ekstutorialcluster`
4. Con el balanceador de carga correcto seleccionado, selecciona la pestaña Descripción.
5. Copia y pega el valor del nombre DNS en la barra de direcciones del navegador.

El «¡Hola, mundo!» La página web aparece en su navegador, lo que indica que la aplicación se ha implementado correctamente.

Paso 9: Realice un cambio en los archivos fuente

En esta sección, realizará un cambio en el `index.html` archivo de su repositorio de código fuente. Este cambio hace que el flujo de trabajo cree una nueva imagen de Docker, la etiquete con un ID de confirmación, la envíe a Amazon ECR y la despliegue en Amazon ECS.

Para cambiar el archivo `index.html`

1. Ve a tu entorno de desarrollo.
2. En la línea de comandos del terminal, cambia a tu repositorio de código fuente:

```
cd /projects/codecatalyst-eks-source-repository
```

3. Obtenga los cambios más recientes en el flujo de trabajo:

```
git pull
```

4. Abra `codecatalyst-eks-source-repository/public-html/index.html`.
5. En la línea 14, cambia el `Hello, World!` texto a `Tutorial complete!`.
6. Agrega, confirma y presiona:

```
git add .  
git commit -m "update index.html title"  
git push
```

La ejecución de un flujo de trabajo se inicia automáticamente.

7. (Opcional) Introduzca:

```
git show HEAD
```

Anote el ID de confirmación del `index.html` cambio. Este ID de confirmación se etiquetará en la imagen de Docker que se implementará en la ejecución del flujo de trabajo que acabas de iniciar.

8. Observe el progreso de la implementación:
 - a. En la CodeCatalyst consola, en el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija `codecatalyst-eks-workflow` ver la última ejecución.
 - c. Elija y BuildBackend haga clic en `DeployToEKS` para ver el progreso de la ejecución del flujo de trabajo.
9. Compruebe que la aplicación se haya actualizado de la siguiente manera:
 - a. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
 - b. A la izquierda, cerca de la parte inferior, selecciona Load Balancers.
 - c. Selecciona el balanceador de carga que se creó como parte de tu implementación de Kubernetes.

- d. Copia y pega el valor del nombre DNS en la barra de direcciones de tu navegador.

¡El «Tutorial completo!» La página web aparece en su navegador, lo que indica que ha implementado correctamente una nueva revisión de la aplicación.

10. (Opcional) En AWS, cambie a la consola Amazon ECR y compruebe que la nueva imagen de Docker esté etiquetada con el ID de confirmación del paso 7 de este procedimiento.

Limpieza

Debe limpiar su entorno para que no se le cobre innecesariamente por los recursos de almacenamiento y cómputo utilizados en este tutorial.

Eliminación

1. Elimine el clúster:

- En la terminal Dev Environment, introduce:

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Donde:

- *us-west-2* se sustituye por tu región.
- *codecatalyst-eks-cluster* se sustituye por el nombre del clúster que ha creado.

Después de 5 a 10 minutos, se eliminan el clúster y los recursos asociados, incluidos, entre otros, las AWS CloudFormation pilas, los grupos de nodos (en Amazon EC2) y los balanceadores de carga.

Important

Si el `eksctl delete cluster` comando no funciona, es posible que deba actualizar sus AWS credenciales o sus credenciales. `kubectl` Si no está seguro de qué credenciales actualizar, actualice primero las AWS credenciales. Para actualizar sus AWS credenciales, consulte [¿Cómo puedo corregir los errores «No se pueden localizar](#)

[las credenciales» y ExpiredToken «»?](#). Para actualizar sus kubectl credenciales, consulte [¿Cómo puedo corregir los errores «No se puede conectar al servidor»?](#).

2. En la AWS consola, limpie de la siguiente manera:
 1. En Amazon ECR, codecatalyst-eks-image-repo elimínelo.
 2. En el Centro de identidades de IAM, elimine:
 - a. codecatalyst-eks-user
 - b. codecatalyst-eks-permission-set
 3. En IAM, elimine:
 - codecatalyst-eks-build-role
 - codecatalyst-eks-deploy-role
 - codecatalyst-eks-build-policy
 - codecatalyst-eks-deploy-policy
3. En la CodeCatalyst consola, limpie de la siguiente manera:
 1. Eliminarcodecatalyst-eks-workflow.
 2. Eliminarcodecatalyst-eks-environment.
 3. Eliminarcodecatalyst-eks-source-repository.
 4. Elimine su entorno de desarrollo.
 5. Eliminecodecatalyst-eks-project.

En este tutorial, aprendió a implementar una aplicación en un servicio de Amazon EKS mediante un CodeCatalyst flujo de trabajo y una acción de implementación en un clúster de Kubernetes.

Añadir la acción «Desplegar AWS CloudFormation pila»

Tip

Para ver un tutorial que muestra cómo utilizar la acción Implementar AWS CloudFormation pila, consulte [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#).

En esta sección se describe cómo añadir la acción Implementar AWS CloudFormation pila a su flujo de trabajo. La acción crea una CloudFormation pila AWS basada en una plantilla que usted proporcione. La plantilla puede ser:

- AWS CloudFormation plantilla: para obtener más información, consulte [Trabajar con AWS CloudFormation plantillas](#).
- AWS SAM plantilla: para obtener más información, consulte la [especificación AWS Serverless Application Model \(AWS SAM\)](#).

Note

Para usar una AWS SAM plantilla, primero debe empaquetar la AWS SAM aplicación mediante la [sam package](#) operación. Para ver un tutorial que te muestra cómo empaquetar este paquete automáticamente como parte de un CodeCatalyst flujo de trabajo de Amazon, consulta [Tutorial: Implemente una aplicación sin servidor mediante AWS CloudFormation](#).

Si la pila ya existe, la acción ejecuta la CloudFormation [CreateChangeSet](#) operación y, a continuación, la [ExecuteChangeSet](#) operación. A continuación, la acción espera a que se implementen los cambios y se marca como correcta o fallida, en función de los resultados.

Usa la acción Implementar AWS CloudFormation pila si ya tienes una AWS SAM plantilla AWS CloudFormation o plantilla que contenga los recursos que te gustaría implementar, o si planeas generar una automáticamente como parte de una [acción de creación](#) de flujo de trabajo con herramientas como AWS SAM y [AWS Cloud Development Kit \(AWS CDK\)](#).

No hay restricciones en cuanto a la plantilla que puedes usar: puedes crearla CloudFormation o AWS SAM usarla con la acción Implementar AWS CloudFormation pila.

Visual

Para añadir la acción «Desplegar AWS CloudFormation pila», utiliza el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busca la acción Implementar AWS CloudFormation pila y realiza una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Deploy AWS CloudFormation stack. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «Implementar AWS CloudFormation pila»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «Desplegar AWS CloudFormation pila» mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busca la acción Implementar AWS CloudFormation pila y realiza una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Deploy AWS CloudFormation stack. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acción «Implementar AWS CloudFormation pila»](#)
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Variables generadas por la acción «Implementar AWS CloudFormation pila»

Cuando se ejecuta la acción Implementar una AWS CloudFormation pila, genera variables que puede usar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «Desplegar AWS CloudFormation pila»](#) la [Lista de variables predefinidas](#).

Definición de la acción «Desplegar AWS CloudFormation pila»

La acción Implementar AWS CloudFormation pila se define como un conjunto de propiedades de YAML dentro del archivo de definición de flujo de trabajo. Para obtener información sobre estas

propiedades, consulte [Referencia de acción «Implementar AWS CloudFormation pila»](#) la [Referencia de definición de flujo de trabajo](#).

Añadir la acción «Implementar en Amazon ECS»

Tip

Para ver un tutorial que muestra cómo utilizar la acción Implementar en Amazon ECS, consulte [Tutorial: Implementación de una aplicación en Amazon ECS](#).

Tip

Para ver un ejemplo práctico de la acción Implementar en Amazon ECS, cree un proyecto con la API de Node.js AWS Fargate o con la API de Java con AWS Fargate blueprint. Para obtener más información, consulte [Crear un proyecto con un plano](#).

En esta sección se describe cómo añadir la acción Implementar en Amazon ECS a su flujo de trabajo. Esta acción registra un archivo de [definición de tareas](#) que usted proporciona. [Tras el registro, el servicio Amazon ECS que se ejecuta en el clúster de Amazon ECS crea una instancia de la definición de la tarea](#). «Crear una instancia de una definición de tarea» equivale a implementar una aplicación en Amazon ECS.

Para utilizar esta acción, debe tener preparado un archivo de definición de clúster, servicio y tarea de Amazon ECS existente.

Para obtener más información sobre Amazon ECS, consulte la Guía para desarrolladores de Amazon Elastic Container Service.

Visual

Para añadir la acción «Implementar en Amazon ECS» mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción Implementar en Amazon ECS y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Implementar en Amazon ECS. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «Implementar en Amazon ECS»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «Implementar en Amazon ECS» mediante el editor YAML

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.

6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción Implementar en Amazon ECS y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Implementar en Amazon ECS. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acción «Implementar en Amazon ECS»](#)
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Variables generadas por la acción «Implementar en Amazon ECS»

Cuando se ejecuta la acción Deploy to Amazon ECS, genera variables que puede utilizar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «Implementar en Amazon ECS»](#) la [Lista de variables predefinidas](#).

Definición de la acción «Implementar en Amazon ECS»

La acción Implementar en Amazon ECS se define como un conjunto de propiedades YAML dentro del archivo de definición de flujo de trabajo. Para obtener información sobre estas propiedades, consulte [Referencia de acción «Implementar en Amazon ECS»](#) la [Referencia de definición de flujo de trabajo](#).

Añadir la acción «Implementar en un clúster de Kubernetes»

Tip

Para ver un tutorial que muestra cómo utilizar la acción Implementar en un clúster de Kubernetes, consulte. [Tutorial: Implementación de una aplicación en Amazon EKS](#)

En esta sección, se describe cómo añadir la acción Implementar en un clúster de Kubernetes a tu flujo de trabajo. Esta acción despliega la aplicación en un clúster de Kubernetes que haya configurado en Amazon Elastic Kubernetes Service (EKS) mediante uno o varios archivos de manifiesto de Kubernetes que usted proporcione. Para [deployment.yaml Tutorial: Implementación de una aplicación en Amazon EKS](#) ver un ejemplo de manifiesto, consulte en.

[Para obtener más información sobre Kubernetes, consulta la documentación de Kubernetes.](#)

Para obtener más información sobre Amazon EKS, consulte [¿Qué es Amazon EKS?](#) en la Guía del usuario de Amazon EKS.

Cómo funcionan

La implementación en el clúster de Kubernetes funciona de la siguiente manera:

1. En tiempo de ejecución, la acción instala la `kubectl` utilidad Kubernetes en la máquina de CodeCatalyst procesamiento en la que se ejecuta la acción. La acción se configura `kubectl` para apuntar al clúster de Amazon EKS que proporcionó al configurar la acción. A continuación, es necesaria la `kubectl` utilidad para ejecutar el `kubectl apply` comando.
2. La acción ejecuta el `kubectl apply -f my-manifest.yaml` comando, que sigue las instrucciones de `my-manifest.yaml` para implementar la aplicación como un conjunto de contenedores y pods en el clúster configurado. Para obtener más información sobre este comando, consulta el [tema kubectl apply en la documentación](#) de referencia de Kubernetes.

Requisitos previos

Para utilizar esta acción, debe tener preparado lo siguiente:

Tip

Para configurar estos requisitos previos rápidamente, siga las instrucciones que se indican en [Tutorial: Implementación de una aplicación en Amazon EKS](#).

- Un clúster de Kubernetes en Amazon EKS. Para obtener información sobre los clústeres, consulte los [clústeres de Amazon EKS](#) en la Guía del usuario de Amazon EKS.
- Al menos un Dockerfile que describa cómo ensamblar la aplicación en una imagen de Docker. [Para obtener más información sobre Dockerfiles, consulta la referencia de Dockerfile.](#)
- Al menos un archivo de manifiesto de Kubernetes, que se denomina archivo de configuración o configuración en la documentación de Kubernetes. Para obtener más información, consulte [Administrar](#) recursos en la documentación de Kubernetes.
- Una función de IAM que proporciona a la acción de implementación en el clúster de Kubernetes la capacidad de acceder a su clúster de Amazon EKS e interactuar con él. Para obtener más información, consulte el tema [Role](#) en la [Referencia de acción «Implementar en un clúster de Kubernetes»](#).

Tras crear este rol, debe añadirlo a:

- Tu archivo de Kubernetes ConfigMap . Para obtener información sobre cómo añadir un rol a un ConfigMap archivo, consulte [Habilitar el acceso principal de IAM a su clúster](#) en la Guía del usuario de Amazon EKS.
- CodeCatalyst. Para obtener información sobre cómo añadir una función de IAM a CodeCatalyst, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).
- Un CodeCatalyst espacio, un proyecto y un entorno. Tanto el espacio como el entorno deben estar conectados a la AWS cuenta en la que vaya a implementar la aplicación. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#), [Crear un proyecto vacío en Amazon CodeCatalyst](#) y [Trabajo con entornos](#).
- Un repositorio de código fuente compatible con CodeCatalyst. El repositorio almacena los archivos fuente de la aplicación, los Dockerfiles y los manifiestos de Kubernetes. Para obtener más información, consulte [Repositorios de fuentes en CodeCatalyst](#).

Añadir la acción «Implementar en un clúster»

Sigue las siguientes instrucciones para añadir la acción Implementar en un clúster a tu flujo de trabajo.

Visual

Para añadir la acción «Implementar en un clúster de Kubernetes» mediante el editor visual

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busca la acción Implementar en el clúster de Kubernetes y realiza una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Implementar en un clúster de Kubernetes. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «Implementar en un clúster de Kubernetes»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «Implementar en un clúster de Kubernetes» mediante el editor YAML

1. [Abre la consola en https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busca la acción Implementar en el clúster de Kubernetes y realiza una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija Implementar en un clúster de Kubernetes. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acción «Implementar en un clúster de Kubernetes»](#)
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Variables generadas por la acción «Implementar en un clúster de Kubernetes»

Cuando se ejecuta la acción Implementar en un clúster de Kubernetes, genera variables que puedes usar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «Desplegar en un clúster de Kubernetes»](#) la [Lista de variables predefinidas](#)

Definición de la acción «Implementar en un clúster de Kubernetes»

La acción Implementar en un clúster de Kubernetes se define como un conjunto de propiedades YAML incluidas en el archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte en la [Referencia de acción «Implementar en un clúster de Kubernetes»](#) [Referencia de definición de flujo de trabajo](#)

Añadir la acción «AWS CDK desplegar»

En esta sección se describe cómo añadir la acción de AWS CDK despliegue a su flujo de trabajo. La acción de AWS CDK despliegue sintetiza y despliega tu AWS Cloud Development Kit (AWS CDK) aplicación en. AWS Si tu aplicación ya existe en AWS, la acción la actualiza si es necesario.

Para obtener información general sobre cómo escribir aplicaciones con AWS CDK, consulta [¿Qué es AWS CDK?](#) en la Guía para AWS Cloud Development Kit (AWS CDK) desarrolladores.

Temas

- [Cuándo usar esta acción](#)
- [Cómo funcionan](#)
- [Versiones de la CLI de CDK utilizadas por la acción «AWS CDK implementar»](#)
- [¿Cuántas pilas puede desplegar la acción?](#)
- [Requisitos previos](#)
- [Ejemplo de flujo de trabajo](#)
- [Añadir la acción «desplegar»AWS CDK](#)
- [Variables generadas por la acción «AWS CDK desplegar»](#)
- [definición de acción «AWS CDK desplegar»](#)

Cuándo usar esta acción

Utilice esta acción si ha desarrollado una aplicación con el AWS CDK y ahora desea implementarla automáticamente como parte de un flujo de trabajo automatizado de integración y entrega continuas

(CI/CD). Por ejemplo, es posible que desees implementar tu AWS CDK app automáticamente cada vez que alguien fusione una solicitud de cambios relacionada con la fuente de tu AWS CDK aplicación.

Cómo funcionan

La AWS CDK implementación funciona de la siguiente manera:

1. [En tiempo de ejecución, si especificó la versión 1.0.12 o anterior de la acción, la acción descarga la última CLI de CDK \(también denominada kit de herramientas\) AWS CDK a la imagen de compilación. CodeCatalyst](#)

Si especificó la versión 1.0.13 o posterior, la acción viene incluida con una [versión específica de la CLI del CDK](#), por lo que no se produce ninguna descarga.

2. La acción utiliza la CLI de CDK para ejecutar el `cdk deploy` comando. Este comando sintetiza e implementa la aplicación en ella. AWS CDK AWSPara obtener más información sobre este comando, consulte el tema del [AWS CDK kit de herramientas \(comando cdk\)](#) de la Guía para desarrolladores.AWS Cloud Development Kit (AWS CDK)

Versiones de la CLI de CDK utilizadas por la acción «AWS CDK implementar»

En la siguiente tabla se muestra qué versión de la CLI de CDK utilizan de forma predeterminada las distintas versiones de la acción de AWS CDK despliegue.

Note

Es posible que pueda anular la predeterminada. Para obtener más información, consulte [CdkCliVersion](#) en la [referencia de acción «AWS CDK desplegar»](#).

Versión de acción AWS CDK para «desplegar»	AWS CDK Versión CLI
1.0.0 — 1.0.12	más recientes
1.0.13 o posterior	2.99.1

¿Cuántas pilas puede desplegar la acción?

La AWS CDK implementación solo puede implementar una sola pila. Si tu AWS CDK aplicación consta de varias pilas, debes crear una pila principal con pilas anidadas e implementar la principal mediante esta acción.

Requisitos previos

Antes de poder utilizar la acción de AWS CDK despliegue, complete las siguientes tareas:

1. Ten preparada una AWS CDK aplicación. Puedes escribir tu AWS CDK aplicación usando la AWS CDK v1 o la v2, en cualquier lenguaje de programación compatible con AWS CDK. Asegúrate de que los archivos de tu AWS CDK aplicación estén disponibles en:
 - Un [repositorio CodeCatalyst de origen](#), o
 - Un [artefacto CodeCatalyst de salida](#) generado por otra acción del flujo de trabajo
2. Inicie su entorno. AWS Para arrancar, puede:
 - Use uno de los métodos descritos en [Cómo arrancar en la Guía](#) para AWS Cloud Development Kit (AWS CDK) desarrolladores.
 - Utilice la acción de AWS CDK arranque. Puedes añadir esta acción en el mismo flujo de trabajo que tu AWS CDK implementación o en uno diferente. Solo asegúrate de que la acción de arranque se ejecute al menos una vez antes de ejecutar la acción de AWS CDK despliegue para disponer de los recursos necesarios. Para obtener más información sobre la acción de AWS CDK arranque, consulte. [Añadir la acción «AWS CDK bootstrap»](#)

Para obtener más información sobre el arranque, consulte Bootstrapping en la Guía para [desarrolladores](#).AWS Cloud Development Kit (AWS CDK)

Ejemplo de flujo de trabajo

El siguiente ejemplo de flujo de trabajo incluye la acción de AWS CDK despliegue junto con la acción de AWS CDK arranque. El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

Note

El siguiente ejemplo de flujo de trabajo tiene fines ilustrativos y no funcionará sin una configuración adicional. Su objetivo es dar un ejemplo del aspecto que tendría un flujo de trabajo cuando se configura con la acción de AWS CDK despliegue.

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Este repositorio contiene tu AWS CDK aplicación. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de AWS CDK arranque (CDKBootstrap): al activarse, la acción despliega la pila de CDKToolkit arranque en ella. AWS Si la CDKToolkit pila ya existe en el entorno, se actualizará si es necesario; de lo contrario, no ocurrirá nada y la acción se marcará como correcta.
- Una acción de AWS CDK despliegue (AWS CDK Deploy): al finalizar la acción de AWS CDK arranque, la acción de AWS CDK despliegue sintetiza el código de AWS CDK la aplicación en una AWS CloudFormation plantilla y despliega la pila definida en la plantilla. AWS

```
Name: codecatalyst-cdk-deploy-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  CDKBootstrap:
    Identifier: aws/cdk-bootstrap@v1
    Inputs:
      Sources:
        - WorkflowSource
    Environment:
      Name: codecatalyst-cdk-deploy-environment
      Connections:
        - Name: codecatalyst-account-connection
          Role: codecatalyst-cdk-bootstrap-role
    Configuration:
      Region: us-west-2

  CDKDeploy:
```

```
Identifier: aws/cdk-deploy@v1
DependsOn:
  - CDKBootstrap
Environment:
  Name: codecatalyst-cdk-deploy-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-cdk-deploy-role
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  StackName: my-app-stack
  Region: us-west-2
```

Añadir la acción «desplegar»AWS CDK

Sigue las siguientes instrucciones para añadir la acción de AWS CDK despliegue a tu flujo de trabajo.

Requisitos previos

Antes de empezar, asegúrese de haber completado las tareas descritas en [Requisitos previos](#).

Visual

Para añadir la acción «AWS CDK desplegar», utilice el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de AWS CDK despliegue y realice una de las siguientes acciones:

- Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)

- Elija AWS CDK implementar. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [referencia de acción «AWS CDK desplegar»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
 11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

Note

Si la acción de AWS CDK despliegue falla y se produce un `npm install error`, consulta [¿Cómo puedo corregir los errores de «npm install»? para obtener información sobre cómo corregir el error.](#)


YAML

Para añadir la acción «AWS CDK implementar» mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de AWS CDK despliegue y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija AWS CDK implementar. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Descargar para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [referencia de acción «AWS CDK desplegar»](#)
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

 Note

Si la acción de AWS CDK despliegue falla y se produce un `npm install error`, consulta [¿Cómo puedo corregir los errores de «npm install»? para obtener información sobre cómo corregir el error.](#)

Variables generadas por la acción «AWS CDK desplegar»

Cuando se ejecuta la acción de AWS CDK despliegue, genera variables que puede utilizar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [«AWS CDKdesplegar» variables de acción](#) la [Lista de variables predefinidas](#).

definición de acción «AWS CDK desplegar»

La acción de AWS CDK despliegue se define como un conjunto de propiedades YAML dentro del archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte [referencia de acción «AWS CDK desplegar»](#) la [Referencia de definición de flujo de trabajo](#).

Trabajo con implementaciones

En esta sección se describe cómo gestionar, supervisar y configurar las implementaciones en Amazon CodeCatalyst.

Temas

- [Implementación de una aplicación o un recurso](#)
- [Volver a ejecutar una implementación](#)
- [Ver el estado de la implementación, las confirmaciones y las solicitudes de cambios](#)
- [Visualización de los registros de despliegue](#)
- [Configuración de reversiones](#)
- [Mostrar la URL de la aplicación implementada](#)
- [Eliminar un objetivo de despliegue](#)

Implementación de una aplicación o un recurso

Para implementar una aplicación o un recurso mediante Amazon CodeCatalyst, debes usar un CodeCatalyst flujo de trabajo. Este flujo de trabajo debe incluir al menos una acción que pueda implementar algo. Por ejemplo, si quisiera implementar una AWS CloudFormation pila, crearía un flujo de trabajo que incluyera la acción Implementar una AWS CloudFormation pila.

Para obtener más información acerca de los flujos de trabajo, consulte [Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst](#). Para obtener más información sobre las acciones, consulte [Trabajar con acciones](#).

Volver a ejecutar una implementación

Para volver a ejecutar una implementación en Amazon CodeCatalyst, debes volver a ejecutar su flujo de trabajo asociado. Para obtener instrucciones, consulte [Iniciar una ejecución de flujo de trabajo](#).

Ver el estado de la implementación, las confirmaciones y las solicitudes de cambios

Puedes ver la siguiente información sobre una implementación en Amazon CodeCatalyst:

- La actividad de despliegue, incluido el estado del despliegue, la hora de inicio, la hora de finalización, el historial y la duración de los eventos.
- Nombre de la pila Región de AWS, hora de la última actualización y flujos de trabajo asociados.
- Solicitudes de confirmación y extracción.
- Información específica sobre la acción, por ejemplo, CloudFormation eventos y resultados.

[Puede ver la información de despliegue a partir de un flujo de trabajo, un entorno o una acción del flujo de trabajo.](#)

Para ver la información de despliegue a partir de un flujo de trabajo

- Vaya a la ejecución del flujo de trabajo en la que se implementó la aplicación. Para ver instrucciones, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Para ver la información de implementación a partir de un entorno

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
4. Elija el entorno en el que se implementó la pila, por ejemplo. `Production`
5. Seleccione Actividad de implementación para ver el historial de implementación de sus pilas, el estado de las implementaciones (por ejemplo, EXITOSAS o FALLIDAS) y otra información relacionada con la implementación.
6. Seleccione Destino de despliegue para ver información sobre las pilas, los clústeres u otros objetivos implementados en el entorno. Puede ver información como el nombre de la pila, la región, el proveedor y el identificador.

Para ver la información de despliegue a partir de una acción

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.

3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. En el diagrama de flujo de trabajo, elija la acción de flujo de trabajo que implementó la aplicación. Por ejemplo, puede elegir DeployCloudFormationStack.
6. Revise el contenido del panel derecho para obtener información sobre la implementación específica de cada acción.

Visualización de los registros de despliegue

Puedes ver los registros relacionados con acciones de implementación específicas para solucionar problemas en Amazon CodeCatalyst.

Puedes ver los registros partiendo de un [flujo de trabajo](#) o un [entorno](#).

Para ver los registros de una acción de despliegue a partir de un flujo de trabajo

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Seleccione Ejecuciones.
6. Elija la ejecución del flujo de trabajo en la que se desplegó la aplicación.
7. En el diagrama de flujo de trabajo, elija la acción cuyos registros desee ver.
8. Seleccione la pestaña Registros y expanda las secciones para mostrar los mensajes de registro.
9. Para ver más registros, selecciona la pestaña Resumen y, a continuación, selecciona Ver en CloudFormation (si está disponible) para ver allí más registros. Puede que tengas que iniciar sesión en AWS.

Para ver los registros de una acción de despliegue a partir de un entorno

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.

4. Elija el entorno en el que se implementó la aplicación.
5. En la actividad de despliegue, busque la columna ID de ejecución del flujo de trabajo y elija la ejecución del flujo de trabajo que desplegó la pila.
6. En el diagrama de flujo de trabajo, elige la acción cuyos registros quieres ver.
7. Seleccione la pestaña Registros y expanda las secciones para mostrar los mensajes de registro.
8. Para ver más registros, selecciona la pestaña Resumen y, a continuación, selecciona Ver en CloudFormation (si está disponible) para ver allí más registros. Puede que tengas que iniciar sesión en AWS.

Configuración de reversiones

De forma predeterminada, si la acción Implementar AWS CloudFormation pila falla, la pila se revertirá al último estado estable conocido. AWS CloudFormation Puede cambiar el comportamiento para que las reversiones se produzcan no solo cuando la acción falle, sino también cuando se produzca una CloudWatch alarma de Amazon específica. Para obtener más información sobre CloudWatch las alarmas, consulta [Uso de CloudWatch alarmas de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

También puedes cambiar el comportamiento predeterminado para que CloudFormation no se revierta la pila si la acción falla.

Usa las siguientes instrucciones para configurar las reversiones.

Note

No puede iniciar una reversión manualmente.

Visual

Antes de empezar

1. Asegúrese de tener un [flujo de trabajo](#) que incluya una acción de implementación de AWS CloudFormation pila que funcione. Para obtener más información, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).
2. En el rol especificado en el campo Función de pila: opcional de la acción Implementar AWS CloudFormation pila, asegúrate de incluir el CloudWatchFullAccesspermiso. Para obtener


información sobre cómo crear este rol con los permisos adecuados, consulte [Paso 2: Crear AWS roles](#).

Para configurar las alarmas de reversión para la acción «Implementar AWS CloudFormation pila»

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de un flujo de trabajo que incluya la acción Implementar AWS CloudFormation pila. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. Elige la acción de despliegue de la AWS CloudFormation pila.
8. En el panel de detalles, selecciona Configuración.
9. En la parte inferior, expande Avanzado.
10. En Supervisar los ARN de las alarmas, selecciona Añadir alarma.
11. Introduce la información en los siguientes campos.

- ARN de alarma

Especifique el nombre de recurso de Amazon (ARN) de una CloudWatch alarma de Amazon para utilizarla como activador de reversión. Por ejemplo, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Puede tener un máximo de cinco activadores de reversión.

 Note

Si especificas un CloudWatch ARN de alarma, también tendrás que configurar permisos adicionales para permitir el acceso de la acción. CloudWatch Para obtener más información, consulte [Configuración de reversiones](#).

- Tiempo de monitoreo

Especifique un período de tiempo, de 0 a 180 minutos, durante el cual se CloudFormation supervisarán las alarmas especificadas. La supervisión comienza después de que se

hayan desplegado todos los recursos de la pila. Si la alarma se produce dentro del tiempo de supervisión especificado, la implementación falla y CloudFormation revierte toda la operación de apilamiento.

Predeterminado: 0. CloudFormation solo supervisa las alarmas mientras se despliegan los recursos de la pila, no después.

YAML

Para configurar los activadores de reversión para la acción «Implementar AWS CloudFormation pila»

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de un flujo de trabajo que incluya la acción Implementar AWS CloudFormation pila. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Añada las `monitor-timeout-in-minutes` propiedades `monitor-alarm-arns` y en el código YAML para añadir activadores de reversión. Para obtener una explicación de cada propiedad, consulte [Referencia de acción «Implementar AWS CloudFormation pila»](#)
8. En el rol especificado en la `role-arn` propiedad de la acción Implementar AWS CloudFormation pila, asegúrese de incluir el `CloudWatchFullAccess` permiso. Para obtener información sobre cómo crear este rol con los permisos adecuados, consulte [Paso 2: Crear AWS roles](#).

Visual

Para desactivar las reversiones de la acción «Implementar AWS CloudFormation pila»

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el nombre de un flujo de trabajo que incluya la acción Implementar AWS CloudFormation pila. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. Elige la acción de despliegue de la AWS CloudFormation pila.
8. En el panel de detalles, selecciona Configuración.
9. En la parte inferior, expande Avanzado.
10. Activa Desactivar la reversión.

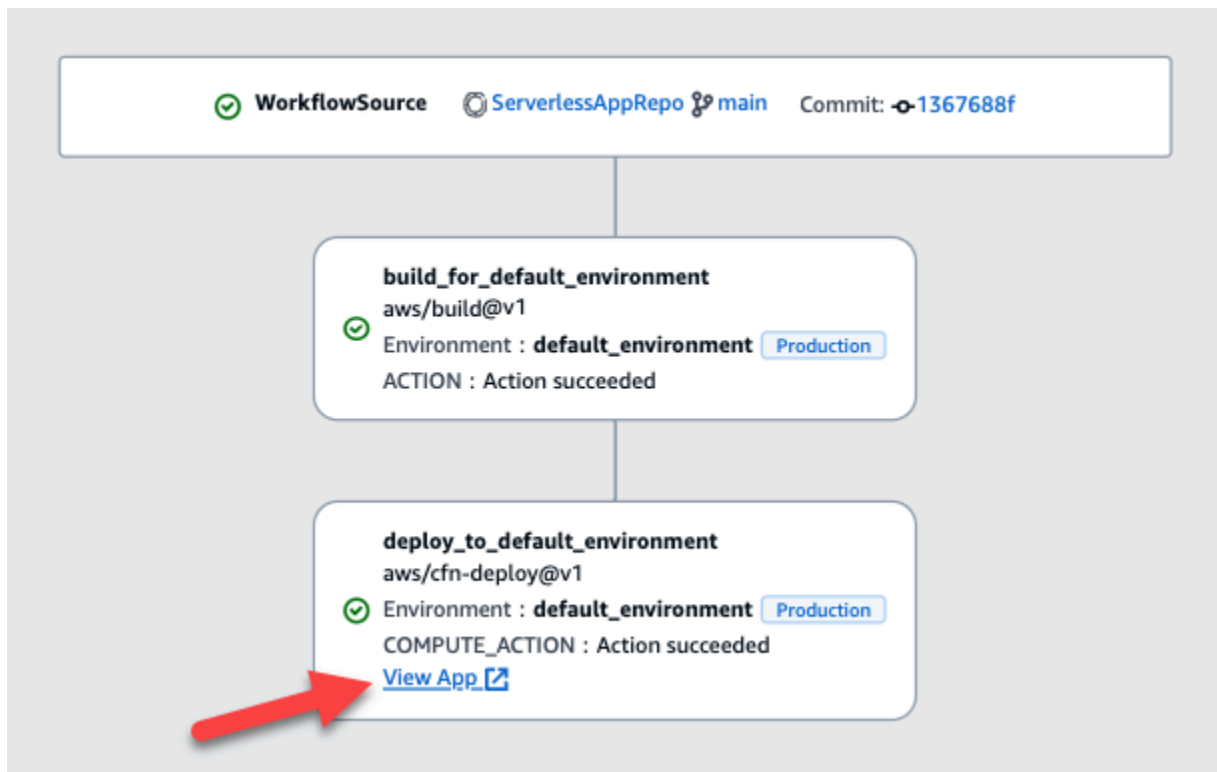
YAML

Para desactivar las reversiones de la acción «Desplegar AWS CloudFormation pila»

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de un flujo de trabajo que incluya la acción Implementar AWS CloudFormation pila. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Agrega la `disable-rollback: 1` propiedad en el código YAML para detener las reversiones. Para obtener una explicación de esta propiedad, consulte. [Referencia de acción «Implementar AWS CloudFormation pila»](#)

Mostrar la URL de la aplicación implementada

Si tu flujo de trabajo implementa una aplicación, puedes configurar Amazon CodeCatalyst para que muestre la URL de la aplicación como un enlace en el que se puede hacer clic. Este enlace aparece en la CodeCatalyst consola, dentro de la acción que lo implementó. El siguiente diagrama de flujo de trabajo muestra la URL de View App que aparece en la parte inferior de una acción.



Al hacer clic en esta URL en la CodeCatalyst consola, puede verificar rápidamente la implementación de la aplicación.

Note

La URL de la aplicación no es compatible con la acción Implementar en Amazon ECS.

Para habilitar esta función, añada una variable de salida a su acción con un nombre que contenga `appurl` o `endpointurl`. Puede usar un nombre con o sin un guión (-), un guión bajo (_) o un espacio () que lo unan. La cadena no distingue entre mayúsculas y minúsculas. Establezca el valor de la variable en la `https` URL `http` o la de la aplicación implementada.

Note

Si está actualizando una variable de salida existente para incluir la `endpoint url` cadena `oapp url`, actualice todas las referencias a esta variable para usar el nombre de la nueva variable.

Para ver los pasos detallados, consulte uno de los siguientes procedimientos:

- [Para mostrar la URL de la aplicación en la acción «AWS CDK implementar»](#)
- [Para mostrar la URL de la aplicación en la acción «Implementar AWS CloudFormation pila»](#)
- [Para mostrar la URL de la aplicación en todas las demás acciones](#)

Cuando haya terminado de configurar la URL, compruebe que aparece según lo esperado siguiendo estas instrucciones:

- [Para comprobar que se agregó la URL de la aplicación](#)

Para mostrar la URL de la aplicación en la acción «AWS CDK implementar»

1. Si utilizas la acción de AWS CDK despliegue, añade una `CfnOutput` construcción (que es un par clave-valor) en el código de tu AWS CDK aplicación:
 - El nombre de la clave debe contener `appurl`, `endpointurl`, con o sin un guión (-), un guión bajo (_) o un espacio (). La cadena no distingue entre mayúsculas y minúsculas.
 - El valor debe ser la `https` URL `http` o la URL de la aplicación implementada.

Por ejemplo, el AWS CDK código podría tener este aspecto:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'APP-URL', {
      value: https://mycompany.myapp.com,
      description: 'The URL of the deployed application',
      exportName: 'myApp',
    });
    ...
  }
}
```

Para obtener más información sobre la CfnOutput construcción, consulta la [interfaz CfnOutputProps](#) en la referencia de la AWS Cloud Development Kit (AWS CDK) API.

2. Guarda y confirma tu código.
3. Continúe en [Para comprobar que se agregó la URL de la aplicación](#).

Para mostrar la URL de la aplicación en la acción «Implementar AWS CloudFormation pila»

1. Si utilizas la acción Implementar AWS CloudFormation pila, añade un resultado a la Outputs sección de tu CloudFormation plantilla o AWS SAM plantilla con estas características:
 - La clave (también denominada identificador lógico) debe contener o `appurlendpointurl`, con o sin un guión (-), un guión bajo (_) o un espacio (). La cadena no distingue entre mayúsculas y minúsculas.
 - El valor debe ser la `https` URL `http` o la URL de la aplicación implementada.

Por ejemplo, la CloudFormation plantilla podría tener este aspecto:

```
"Outputs" : {  
  "APP-URL" : {  
    "Description" : "The URL of the deployed app",  
    "Value" : "https://mycompany.myapp.com",  
    "Export" : {  
      "Name" : "My App"  
    }  
  }  
}
```

Para obtener más información sobre CloudFormation las salidas, consulte [las salidas](#) en la Guía del AWS CloudFormation usuario.

2. Guarde y confirme el código.
3. Continúe en [Para comprobar que se agregó la URL de la aplicación](#).

Para mostrar la URL de la aplicación en todas las demás acciones

Si utilizas otra acción para implementar la aplicación, como la acción de creación o GitHub las acciones, haz lo siguiente para que se muestre la URL de la aplicación.

1. Defina una variable de entorno en la Steps sección Inputs o de la acción del archivo de definición del flujo de trabajo. La variable debe tener las siguientes características:
 - nameDebe contenerappurl, oendpointurl, con o sin un guión (-), un guión bajo (_) o un espacio (). La cadena no distingue entre mayúsculas y minúsculas.
 - El valor debe ser la https URL http o la URL de la aplicación implementada.

Por ejemplo, una acción de compilación podría tener este aspecto:

```
Build-action:
  Identifier: aws/build@v1
  Inputs:
    Variables:
      - Name: APP-URL
        Value: https://mycompany.myapp.com
```

... o esto:

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: APP-URL=https://mycompany.myapp.com
```

Para obtener más información sobre la definición de variables de entorno, consulte [Definir una variable](#).

2. Exporte la variable.

Por ejemplo, la acción de construcción podría tener este aspecto:

```
Build-action:
  ...
  Outputs:
    Variables:
      - APP-URL
```

Para obtener información sobre la exportación de variables, consulte [Exportar una variable para que otras acciones puedan utilizarla](#).

3. (Opcional) Elija Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
4. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.
5. Continúe en [Para comprobar que se agregó la URL de la aplicación](#).

Para comprobar que se agregó la URL de la aplicación

- Inicie una ejecución de flujo de trabajo, si no se ha iniciado automáticamente. La nueva ejecución debe mostrar la URL de la aplicación como un enlace en el que se pueda hacer clic en el diagrama de flujo de trabajo. Para obtener más información sobre el inicio de las ejecuciones, consulte [Iniciar una ejecución de flujo de trabajo](#).

Eliminar un objetivo de despliegue

Puede eliminar un objetivo de despliegue, como un clúster o una AWS CloudFormation pila de Amazon ECS, de la página Entornos de la CodeCatalyst consola.

Important

Al eliminar un objetivo de despliegue, se elimina de la CodeCatalyst consola, pero permanece disponible en el AWS servicio que lo aloja (si aún existe).

Considere la posibilidad de eliminar un objetivo de despliegue si el objetivo se ha quedado obsoleto. CodeCatalyst Los objetivos pueden quedar obsoletos si:

- Eliminó el flujo de trabajo que se implementó en el destino.
- Cambiaste la pila o el clúster en el que vas a realizar la implementación.
- Eliminó la pila o el clúster del CloudFormation servicio Amazon ECS de la AWS consola.

Para eliminar un objetivo de despliegue

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
4. Elija el nombre del entorno que contiene el destino de despliegue que desea eliminar. Para obtener información sobre los entornos, consulte [Trabajo con entornos](#).

5. Seleccione la pestaña Objetivos de despliegue.
6. Seleccione el botón de radio situado junto al objetivo de despliegue que desee eliminar.
7. Elija Eliminar.

El objetivo se elimina de la página.

Trabajar con flujos de trabajo

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Temas

- [Crear, editar y eliminar un flujo de trabajo](#)
- [Visualización del estado del flujo de trabajo](#)
- [Trabajar con acciones](#)
- [Trabajar con artefactos](#)
- [Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución](#)
- [Trabajo con entornos](#)
- [Trabajar con el almacenamiento en caché de archivos](#)
- [Trabajar con paquetes](#)
- [Trabajar con corridas](#)
- [Trabajando con secretos](#)

- [Trabajar con fuentes](#)
- [Trabajar con factores desencadenantes](#)
- [Trabajar con variables](#)

Crear, editar y eliminar un flujo de trabajo

Utilice los siguientes procedimientos para crear, editar y eliminar un flujo de trabajo.

Visual

Para crear un flujo de trabajo mediante el editor visual


1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Seleccione Crear flujo de trabajo.

Aparece el cuadro de diálogo Crear flujo de trabajo.

5. En el campo Repositorio de origen, elija un repositorio de origen en el que residirá el archivo de definición del flujo de trabajo. El archivo se almacenará en la `~/codecatalyst/workflows/` carpeta del repositorio elegido. Si no existe ningún repositorio de origen, [cree uno](#).
6. En el campo Sucursal, elija una rama en la que residirá el archivo de definición del flujo de trabajo.
7. Seleccione Crear.

Amazon CodeCatalyst guarda la información del repositorio y la sucursal en la memoria, pero el flujo de trabajo aún no está comprometido.

8. Elija Visual.
9. Cree el flujo de trabajo:
 - a. (Opcional) En el diagrama de flujo de trabajo, selecciona las casillas Origen y Activadores. Aparece un panel de activadores. Seleccione Añadir disparador para añadir un disparador. Para obtener más información, consulte [Añadir un gatillo para pulsar, tirar o programar](#).
 - b. Selecciona + Acciones (arriba a la izquierda). Aparece el catálogo de acciones.

- c. Elija el signo más (+) dentro de una acción para añadirla al flujo de trabajo. Utilice el panel de la derecha para configurar la acción. Para obtener más información, consulte [Añadir una acción](#).
 - d. (Opcional) Elija las propiedades del flujo de trabajo (arriba a la derecha). Aparece un panel de propiedades del flujo de trabajo. Configure el nombre del flujo de trabajo, el modo de ejecución y el cálculo. Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#) y [Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución](#).
10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 11. Elija Confirmar y, en el cuadro de diálogo Confirmar flujo de trabajo, haga lo siguiente:
 - a. En el nombre del archivo de flujo de trabajo, deje el nombre predeterminado o introduzca el suyo propio.
 - b. En Confirmar mensaje, deja el mensaje predeterminado o introduce el tuyo propio.
 - c. En Repositorio y Sucursal, elija el repositorio y la rama de origen para el archivo de definición del flujo de trabajo. Estos campos deben estar configurados en el repositorio y la rama que especificó anteriormente en el cuadro de diálogo Crear flujo de trabajo. Si lo deseas, puedes cambiar el repositorio y la rama ahora.
-  **Note**

Tras archivar tu archivo de definición de flujo de trabajo, no se podrá asociar a otro repositorio o rama, así que asegúrate de elegirlos con cuidado.
- d. Elija Confirmar para confirmar el archivo de definición del flujo de trabajo.

YAML

Para crear un flujo de trabajo mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Seleccione Crear flujo de trabajo.


Aparece el cuadro de diálogo Crear flujo de trabajo.

5. En el campo Repositorio de origen, elija un repositorio de origen en el que residirá el archivo de definición del flujo de trabajo. El archivo se almacenará en la `~/ .codecatalyst/ workflows/` carpeta del repositorio elegido. Si no existe ningún repositorio de origen, [cree uno](#).
6. En el campo Sucursal, elija una rama en la que residirá el archivo de definición del flujo de trabajo.
7. Seleccione Crear.

Amazon CodeCatalyst guarda la información del repositorio y la sucursal en la memoria, pero el flujo de trabajo aún no está comprometido.

8. Elige YAML.
9. Cree el flujo de trabajo:
 - a. (Opcional) Agrega un disparador al código YAML. Para obtener más información, consulte [Añadir un gatillo para pulsar, tirar o programar](#).
 - b. Selecciona + Acciones (arriba a la izquierda). Aparece el catálogo de acciones.
 - c. Elija el signo más (+) dentro de una acción para añadirla al flujo de trabajo. Utilice el panel de la derecha para configurar la acción. Para obtener más información, consulte [Añadir una acción](#).
 - d. (Opcional) Elija las propiedades del flujo de trabajo (arriba a la derecha). Aparece un panel de propiedades del flujo de trabajo. Configure el nombre del flujo de trabajo, el modo de ejecución y el proceso. Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#) y [Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución](#).
10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Elija Confirmar y, en el cuadro de diálogo Confirmar flujo de trabajo, haga lo siguiente:
 - a. En el nombre del archivo de flujo de trabajo, deje el nombre predeterminado o introduzca el suyo propio.
 - b. En Confirmar mensaje, deja el mensaje predeterminado o introduce el tuyo propio.
 - c. En Repositorio y Sucursal, elija el repositorio y la rama de origen para el archivo de definición del flujo de trabajo. Estos campos deben estar configurados en el repositorio y

la rama que especificó anteriormente en el cuadro de diálogo Crear flujo de trabajo. Si lo deseas, puedes cambiar el repositorio y la rama ahora.

 Note

Tras archivar tu archivo de definición de flujo de trabajo, no se podrá asociar a otro repositorio o rama, así que asegúrate de elegirlos con cuidado.

- d. Elija Confirmar para confirmar el archivo de definición del flujo de trabajo.

Para editar un flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre del flujo de trabajo que desee editar. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Edite el flujo de trabajo. Elige Visual para usar el editor visual o YAML para usar el editor YAML. Puede hacer lo siguiente:
 - a. Edita los activadores. Para obtener más información, consulte [Trabajar con factores desencadenantes](#).
 - b. Edita acciones. Para obtener más información, consulte [Trabajar con acciones](#).
7. Edite propiedades adicionales del flujo de trabajo. Elija las propiedades del flujo de trabajo (arriba a la derecha) para cambiar el nombre del flujo de trabajo, el modo de ejecución o el proceso. Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#) y [Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución](#).

Para eliminar un flujo de trabajo

Al eliminar un flujo de trabajo, se elimina el archivo de definición del flujo de trabajo y las ejecuciones asociadas. Todos AWS los servicios desplegados a partir del flujo de trabajo eliminado deben eliminarse manualmente o se seguirán cobrando por esos servicios.

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre del flujo de trabajo que desee eliminar. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Eliminar.
6. Elija Confirmar.

Visualización del estado del flujo de trabajo

Es posible que desee ver el estado de un flujo de trabajo para ver si hay algún problema de configuración del flujo de trabajo que deba solucionar o para solucionar problemas de ejecución que no se inician. CodeCatalyst evalúa el estado del flujo de trabajo cada vez que crea o actualiza el archivo de [definición del flujo de trabajo subyacente del flujo](#) de trabajo.

Note

También puede ver el estado de ejecución del flujo de trabajo, que es diferente del estado del flujo de trabajo. Para obtener más información, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Un flujo de trabajo puede tener uno de los siguientes estados:

- Válido: [el flujo de trabajo se puede ejecutar y se puede activar mediante activadores](#).

Para que un flujo de trabajo se marque como válido, se deben cumplir las dos condiciones siguientes:

- El archivo de definición del flujo de trabajo debe ser válido.
- El flujo de trabajo no debe tener activadores, ni activadores push, o tener un activador push que se ejecute utilizando los archivos de la rama actual. Para obtener más información, consulte [Consideraciones sobre los factores desencadenantes al ramificar](#).
- No válido: el archivo de definición del flujo de trabajo no es válido. El flujo de trabajo no se puede ejecutar de forma manual ni automática mediante activadores. *Los flujos de trabajo que*

*no son válidos aparecen con una **definición de flujo de trabajo con un mensaje de error** (o similar) en la CodeCatalyst consola.*

Para que un flujo de trabajo se marque como no válido, se debe cumplir la siguiente condición:

- El archivo de definición del flujo de trabajo debe estar mal configurado.

Para corregir un archivo de definición de flujo de trabajo mal configurado, consulte [¿Cómo puedo corregir los errores «La definición de flujo de trabajo no tiene errores»?](#)

- Inactivo: la definición del flujo de trabajo es válida, pero no se puede ejecutar de forma manual ni automática mediante activadores.

Para que un flujo de trabajo se marque como inactivo, se deben cumplir las dos condiciones siguientes:

- El archivo de definición del flujo de trabajo debe ser válido.
- El archivo de definición del flujo de trabajo debe incluir un disparador automático que especifique una rama diferente de la rama en la que se encuentra actualmente el archivo de definición del flujo de trabajo. Para obtener más información, consulte [Consideraciones sobre los factores desencadenantes al ramificar](#).

Para cambiar un flujo de trabajo de inactivo a activo, consulte [¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?](#)

Note

Si el flujo de trabajo especifica un recurso que eliminás más adelante (por ejemplo, un repositorio de paquetes), CodeCatalyst no detectará este cambio y seguirá marcando el flujo de trabajo como válido. Estos tipos de problemas se detectarán cuando se ejecute el flujo de trabajo.

Para ver el estado de un flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Busque el flujo de trabajo cuyo estado desee ver. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por nombre del flujo de trabajo.

El estado se muestra con el flujo de trabajo en la lista.

5. (Opcional) Elija el nombre del flujo de trabajo y busque el campo de definición del flujo de trabajo. Muestra el estado del flujo de trabajo.

Trabajar con acciones

Una acción es el componente principal de un flujo de trabajo y define una unidad lógica de trabajo que se debe realizar durante la ejecución de un flujo de trabajo. Normalmente, un flujo de trabajo incluye varias acciones que se ejecutan de forma secuencial o paralela en función de cómo las haya configurado.

Temas

- [Tipos de acción](#)
- [Añadir, configurar y eliminar una acción](#)
- [Desarrollo de una acción personalizada](#)
- [Agrupación de acciones en grupos de acciones](#)
- [Configurar acciones para que dependan de otras acciones](#)
- [Visualización del código fuente de una acción](#)
- [Trabajar con versiones de acción](#)

Tipos de acción

Dentro de un CodeCatalyst flujo de trabajo de Amazon, puedes usar los siguientes tipos de acciones.

Tipos de acción

- [CodeCatalyst acciones](#)
- [CodeCatalyst Acciones de Labs](#)
- [GitHub Acciones](#)
- [Acciones de terceros](#)

CodeCatalyst acciones

Una CodeCatalyst acción es una acción creada, mantenida y totalmente respaldada por el equipo de CodeCatalyst desarrollo.

Existen CodeCatalyst acciones para crear, probar e implementar aplicaciones, así como para realizar diversas tareas, como la invocación de una función. AWS Lambda

Están disponibles las siguientes acciones: CodeCatalyst

- Build

Esta acción crea tus artefactos y ejecuta tus pruebas unitarias en un contenedor de Docker. Para obtener más información, consulte [Añadir la acción de construcción](#).

- Pruebas

Esta acción ejecuta pruebas de integración y de sistema con la aplicación o los artefactos. Para obtener más información, consulte [Añadir la acción de prueba](#).

- Publicación en Amazon S3

Esta acción copia los artefactos de la aplicación en un bucket de Amazon S3. Para obtener más información, consulte [Añadir la acción «Amazon S3 publish»](#).

- AWS CDK bootstrap

Esta acción proporciona los recursos que se AWS CDK necesitan para implementar su aplicación de CDK. Para obtener más información, consulte [Añadir la acción «AWS CDK bootstrap»](#).

- AWS CDK implementar

Esta acción sintetiza e implementa una AWS Cloud Development Kit (AWS CDK) aplicación. Para obtener más información, consulte [Añadir la acción «AWS CDK desplegar»](#).

- AWS Lambda invocar

Esta acción invoca una AWS Lambda función. Para obtener más información, consulte [Añadir la acción «AWS Lambdainvocar»](#).

- Implemente una pila AWS CloudFormation

Esta acción despliega AWS CloudFormation pilas. Para obtener más información, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).

- Implemente en Amazon ECS

Esta acción registra una definición de tarea de Amazon ECS y la despliega en un servicio de Amazon ECS. Para obtener más información, consulte [Añadir la acción «Implementar en Amazon ECS»](#).

- Implemente en un clúster de Kubernetes

Esta acción despliega una aplicación en un clúster de Kubernetes. Para obtener más información, consulte [Añadir la acción «Implementar en un clúster de Kubernetes»](#).

- Renderizar la definición de tareas de Amazon ECS

Esta acción inserta un URI de imagen de contenedor en un archivo JSON de definición de tareas de Amazon ECS y crea un nuevo archivo de definición de tareas. Para obtener más información, consulte [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#).

La documentación sobre las CodeCatalyst acciones está disponible en esta guía y en el archivo readme de cada acción.

Para obtener información sobre las CodeCatalyst acciones disponibles y cómo añadir una a un flujo de trabajo, consulte [Añadir una acción](#).

CodeCatalyst Acciones de Labs

Una acción de CodeCatalyst Labs es una acción que forma parte de Amazon CodeCatalyst Labs, un campo de pruebas para aplicaciones experimentales. CodeCatalyst Las acciones de Labs se han desarrollado para mostrar las integraciones con AWS los servicios.

Están disponibles las siguientes acciones de CodeCatalyst Labs:

- Implementar en AWS Amplify Hosting

Esta acción despliega una aplicación en Amplify Hosting.

- Implemente en AWS App Runner

Esta acción implementa la imagen más reciente en un repositorio de imágenes de origen en App Runner.

- Implemente en Amazon CloudFront y Amazon S3

Esta acción despliega una aplicación en CloudFront Amazon S3.

- Implemente con AWS SAM

Esta acción despliega la aplicación sin servidor con AWS Serverless Application Model (AWS SAM).

- Invalidar Amazon Cache CloudFront

Esta acción invalida la CloudFront memoria caché de un conjunto de rutas determinado.

- Acción de Webhook saliente

Esta acción permite a los usuarios enviar mensajes dentro de un flujo de trabajo a un servidor web arbitrario mediante una solicitud HTTPS.

- Publicar en AWS CodeArtifact

Esta acción publica los paquetes en un CodeArtifact repositorio.

- Publicar en Amazon SNS

Esta acción permite a los usuarios integrarse con Amazon SNS mediante la creación de un tema, la publicación en un tema o la suscripción a un tema.

- Empuje a Amazon ECR

Esta acción crea y publica una imagen de Docker en un repositorio de Amazon Elastic Container Registry (Amazon ECR).

- Escanea con Amazon CodeGuru Security

Esta acción crea un archivo zip con una ruta de código configurada y utiliza la CodeGuru seguridad para ejecutar un escaneo de código.

- Edición comunitaria de Terraform

Esta acción ejecuta la Edición Comunitaria de Terraform plan y apply sus operaciones.

La documentación de las acciones de CodeCatalyst Labs está disponible en el archivo readme de cada acción.

Para obtener información sobre cómo añadir una acción de CodeCatalyst Labs a un flujo de trabajo y ver su archivo readme, consulte. [Añadir una acción](#)

GitHub Acciones

Una GitHub acción se parece mucho a una [CodeCatalyst acción](#), excepto que se desarrolló para usarse con GitHub flujos de trabajo. Para obtener más información sobre GitHub las acciones, consulte la documentación sobre [GitHub las acciones](#).

Puedes usar una GitHub acción junto con CodeCatalyst las acciones nativas en un CodeCatalyst flujo de trabajo.

Para tu comodidad, la CodeCatalyst consola proporciona acceso a varias GitHub acciones populares. También puedes usar cualquier GitHub acción que aparezca en el [GitHub Marketplace](#) (sujeto a algunas limitaciones).

La documentación de GitHub las acciones está disponible en el archivo readme de cada acción.

Para obtener más información, consulte [Añadir GitHub acciones](#).

Acciones de terceros

Una acción de terceros es una acción creada por un proveedor externo y que está disponible en la CodeCatalyst consola. Un ejemplo de acción de un tercero es la acción Mend SCA, creada por Mend.

La documentación sobre las acciones de terceros está disponible en el archivo readme de cada acción. El proveedor externo también puede proporcionar documentación adicional.

Para obtener información sobre cómo añadir una acción de terceros a un flujo de trabajo y ver su archivo readme, consulte [Añadir una acción](#).

Añadir, configurar y eliminar una acción

Elija uno de los siguientes temas para obtener información sobre cómo añadir la acción a un flujo de trabajo y cómo eliminarla.

Temas

- [Añadir una acción](#)
- [Añadir la acción «Amazon S3 publish»](#)
- [Añadir la acción «AWS CDK bootstrap»](#)
- [Añadir la acción «AWS Lambdainvocar»](#)
- [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#)
- [Añadir GitHub acciones](#)

- [Eliminar una acción](#)

Añadir una acción

Siga las instrucciones siguientes para añadir una acción a un flujo de trabajo y, a continuación, configurarla.

Para añadir y configurar una acción

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. En la parte superior izquierda, selecciona + Acciones. Aparece el catálogo de acciones.
7. En la lista desplegable, realice una de las siguientes acciones:
 - Elige Amazon CodeCatalyst para ver [CodeCatalyst](#), [CodeCatalyst Labs](#) o acciones [de terceros](#).
 - CodeCatalyst las acciones tienen una AWS etiqueta de «by».
 - CodeCatalyst Las acciones de Labs tienen la etiqueta by CodeCatalyst Labs.
 - Las acciones de terceros llevan la etiqueta «por **proveedor**», donde «**vendedor**» es el nombre del proveedor externo.
 - Elija GitHubver una [lista seleccionada de GitHub acciones](#).
8. En el catálogo de acciones, busque una acción y, a continuación, realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción a su flujo de trabajo.
 - Elija el nombre de la acción para ver su archivo readme.
9. Configure la acción. Elige Visual para usar el editor visual o YAML para usar el editor YAML. Para obtener instrucciones detalladas, consulta los siguientes enlaces.

Para obtener instrucciones sobre cómo añadir [CodeCatalystacciones](#), consulte:

- [Añadir la acción de construcción](#)

- [Añadir la acción de prueba](#)
- [Añadir la acción «Amazon S3 publish»](#)
- [Añadir la acción «AWS CDK bootstrap»](#)
- [Añadir la acción «AWS CDK desplegar»](#)
- [Añadir la acción «AWS Lambdainvocar»](#)
- [Añadir la acción «Desplegar AWS CloudFormation pila»](#)
- [Añadir la acción «Implementar en Amazon ECS»](#)
- [Añadir la acción «Implementar en un clúster de Kubernetes»](#)
- [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#)

Para obtener instrucciones sobre cómo añadir [acciones de CodeCatalyst Labs](#), consulta:

- La acción es readme. Para encontrar el archivo readme, elige el nombre de la acción en el catálogo de acciones.

Para obtener instrucciones sobre cómo añadir [GitHub acciones](#), consulte:

- [Añadir GitHub acciones](#)

Para obtener instrucciones sobre cómo añadir [acciones de terceros](#), consulta:

- El léame de la acción. Para encontrar el archivo readme, elige el nombre de la acción en el catálogo de acciones.

10. (Opcional) Selecciona Validar para asegurarte de que el código YAML es válido.

11. Selecciona Confirmar para confirmar los cambios.

Añadir la acción «Amazon S3 publish»

En esta sección se describe cómo añadir la acción de publicación de Amazon S3 a su flujo de trabajo. La acción de publicación de Amazon S3 copia los archivos de un directorio de origen a un bucket de Amazon S3. El directorio de origen puede residir en:

- Un [repositorio de origen](#), o
- Un [artefacto de salida](#) generado por otra acción del flujo de trabajo

Temas

- [¿Cuándo usar esta acción](#)
- [Ejemplo de flujo de trabajo](#)
- [Añadir la acción «Amazon S3 publish»](#)
- [Variables generadas por la acción «Amazon S3 publish»](#)
- [Definición de la acción «Amazon S3 publish»](#)

¿Cuándo usar esta acción

Use esta acción si:

- Tiene un flujo de trabajo que genera los archivos que desea almacenar en Amazon S3.

Por ejemplo, es posible que tenga un flujo de trabajo que cree un sitio web estático que desee alojar en Amazon S3. En este caso, su flujo de trabajo incluiría una [acción](#) de compilación para crear el HTML y los archivos auxiliares del sitio, y una acción de publicación de Amazon S3 para copiar los archivos en Amazon S3.

- Tiene un repositorio de origen que contiene los archivos que desea almacenar en Amazon S3.

Por ejemplo, es posible que tenga un repositorio de código fuente con archivos fuente de aplicaciones que desee archivar todas las noches en Amazon S3.

Ejemplo de flujo de trabajo

El siguiente flujo de trabajo de ejemplo incluye la acción de publicación de Amazon S3, junto con una acción de compilación. El flujo de trabajo crea un sitio web de documentación estática y, a continuación, lo publica en Amazon S3, donde está alojado. El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

Note

El siguiente ejemplo de flujo de trabajo tiene fines ilustrativos y solo funcionará con una configuración adicional.

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de compilación (BuildDocs): al activarla, la acción crea un sitio web de documentación estática (mkdocs build) y añade los archivos HTML asociados y los metadatos auxiliares a un artefacto llamado MyDocsSite. Para obtener más información sobre la acción de creación, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#).
- Una acción de publicación de Amazon S3 (PublishToS3): al finalizar la acción de compilación, esta acción copia el sitio del MyDocsSite artefacto a Amazon S3 para su alojamiento.

```
Name: codecatalyst-s3-publish-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildDocs:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: echo BuildDocs started on `date`
        - Run: pip install --upgrade pip
        - Run: pip install mkdocs
        - Run: mkdocs build
        - Run: echo BuildDocs completed on `date`
    Outputs:
      Artifacts:
        - Name: MyDocsSite
          Files:
            - "site/**/*"

  PublishToS3:
    Identifier: aws/s3-publish@v1
    Environment:
      Name: codecatalyst-s3-publish-environment
```

```
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-s3-publish-build-role
Inputs:
  Sources:
    - WorkflowSource
  Artifacts:
    - MyDocsSite
Configuration:
  DestinationBucketName: my-bucket
  SourcePath: /artifacts/PublishToS3/MyDocSite/site
  TargetPath: my/docs/site
```

Añadir la acción «Amazon S3 publish»

Siga las instrucciones siguientes para añadir la acción de publicación de Amazon S3 a su flujo de trabajo.

Visual

Para añadir la acción «Publicar en Amazon S3» mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de publicación de Amazon S3 y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

- Elija Amazon S3 publish. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:

- (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas, Configuración y Salidas, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «Amazon S3 publish»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
 11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «Publicar en Amazon S3» mediante el editor YAML

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
 2. Elija el proyecto.
 3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 5. Elija Editar.
 6. Elige YAML.
 7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
 8. En la lista desplegable, selecciona Amazon CodeCatalyst.
 9. Busque la acción de publicación de Amazon S3 y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
- O bien
- Elija Amazon S3 publish. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:

- (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acción «Amazon S3 publish»](#)
 11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

Variables generadas por la acción «Amazon S3 publish»

Cuando se ejecuta la acción de publicación de Amazon S3, genera variables que puede utilizar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «Amazon S3 publish»](#) la [Lista de variables predefinidas](#).

Definición de la acción «Amazon S3 publish»

La acción de publicación de Amazon S3 se define como un conjunto de propiedades YAML dentro del archivo de definición de flujo de trabajo. Para obtener información sobre estas propiedades, consulte [Referencia de acción «Amazon S3 publish»](#) la [Referencia de definición de flujo de trabajo](#).

Añadir la acción «AWS CDK bootstrap»

En esta sección, se describe cómo añadir la acción de AWS CDK arranque a su flujo de trabajo. [La acción AWS CDK bootstrap aprovisiona una pila de bootstrap en su AWS entorno mediante la plantilla moderna](#). Si ya existe una pila de arranque, la acción la actualiza si es necesario. Contar con una pila de bootstrap AWS es un requisito previo para implementar una AWS CDK aplicación.

Para obtener más información sobre el arranque, consulta el tema sobre el [arranque](#) en la Guía para desarrolladores. AWS Cloud Development Kit (AWS CDK)

Temas

- [Cuándo usar esta acción](#)
- [Cómo funcionan](#)
- [Versiones de CDK CLI utilizadas por la acción «AWS CDK bootstrap»](#)
- [Requisitos previos](#)

- [Ejemplo de flujo de trabajo](#)
- [Añadir la acción «AWS CDK bootstrap»](#)
- [Variables producidas por la acción «AWS CDK bootstrap»](#)
- [Definición de acción «AWS CDK bootstrap»](#)

Cuándo usar esta acción

Use esta acción si tiene un flujo de trabajo que implementa una AWS CDK aplicación y desea implementar (y actualizar, si es necesario) la pila de bootstrap al mismo tiempo. En este caso, añadiría la acción de AWS CDK arranque al mismo flujo de trabajo que el que implementa la aplicación. AWS CDK

No utilices esta acción si se da alguna de las siguientes situaciones:

- Ya ha desplegado una pila de arranque mediante otro mecanismo y desea mantenerla intacta (sin actualizaciones).
- Desea utilizar una [plantilla de arranque personalizada](#), que no es compatible con la acción de AWS CDK arranque.

Cómo funcionan

El AWS CDK bootstrap funciona de la siguiente manera:

1. [En tiempo de ejecución, si especificó la versión 1.0.7 o anterior de la acción, la acción descarga la última CLI de CDK \(también denominada kit de herramientas\) AWS CDK a la imagen de compilación. CodeCatalyst](#)

Si especificó la versión 1.0.8 o posterior, la acción viene incluida con una [versión específica de la CLI del CDK](#), por lo que no se produce ninguna descarga.

2. La acción utiliza la CLI de CDK para ejecutar el `cdk bootstrap` comando. Este comando realiza las tareas de arranque descritas en el tema de [arranque de la Guía para desarrolladores](#). AWS Cloud Development Kit (AWS CDK)

Versiones de CDK CLI utilizadas por la acción «AWS CDK bootstrap»

La siguiente tabla muestra qué versión de la CLI de CDK utilizan de forma predeterminada las diferentes versiones de la acción de AWS CDK arranque.

Note

Es posible que pueda anular la predeterminada. Para obtener más información, consulte [CdkCliVersion](#) en la [referencia de acción «AWS CDKbootstrap»](#).

Versión de AWS CDK acción «bootstrap»	AWS CDK Versión CLI
1.0.0 — 1.0.7	más recientes
1.0.8 o posterior	2.99.1

Requisitos previos

Antes de poder utilizar la acción de AWS CDK arranque, asegúrate de tener una AWS CDK aplicación lista. La acción de arranque sintetizará la AWS CDK aplicación antes de arrancarla. Puede escribir su aplicación en cualquier lenguaje de programación compatible con AWS CDK

Asegúrate de que los archivos de tu AWS CDK aplicación estén disponibles en:

- Un [repositorio CodeCatalyst de origen](#), o
- Un [artefacto CodeCatalyst de salida](#) generado por otra acción del flujo de trabajo

Ejemplo de flujo de trabajo

Consulte [Ejemplo de flujo de trabajo](#) en el [Añadir la acción «AWS CDK desplegar»](#) para ver un flujo de trabajo que incluya la acción de AWS CDK arranque.

Añadir la acción «AWS CDK bootstrap»

Sigue las siguientes instrucciones para añadir la acción de AWS CDK arranque a tu flujo de trabajo.

Requisitos previos

Antes de empezar, asegúrese de haber completado las tareas descritas en [Requisitos previos](#).

Visual

Para añadir la acción «AWS CDK bootstrap», utilice el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busca la acción AWS CDK bootstrap y realiza una de las siguientes acciones:

- Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)

- Elija AWS CDK bootstrap. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas, Configuración y Salidas, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [referencia de acción «AWS CDKbootstrap»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
 11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

Note

Si la acción de AWS CDK arranque falla y se produce un `npm install error`, consulte [¿Cómo puedo corregir los errores de «npm install»? para obtener información sobre cómo corregir el error.](#)

YAML

Para añadir la acción «AWS CDK bootstrap» mediante el editor YAML

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de AWS CDK arranque y seleccione + para añadirla al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [referencia de acción «AWS CDKbootstrap»](#)
11. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Selecciona Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

Note

Si la acción de AWS CDK arranque falla y se produce un `npm install error`, consulte [¿Cómo puedo corregir los errores de «npm install»? para obtener información sobre cómo corregir el error.](#)

Variables producidas por la acción «AWS CDK bootstrap»

Cuando se ejecuta la acción de AWS CDK arranque, genera variables que puede utilizar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «AWS CDKbootstrap»](#) la [Lista de variables predefinidas](#).

Definición de acción «AWS CDK bootstrap»

La acción de AWS CDK arranque se define como un conjunto de propiedades YAML incluidas en el archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte [referencia de acción «AWS CDKbootstrap»](#) la [Referencia de definición de flujo de trabajo](#)

Añadir la acción «AWS Lambdainvocar»

En esta sección, se describe cómo añadir la acción de AWS Lambdainvocación a su flujo de trabajo. La acción de AWS Lambdainvocación invoca una función Lambda que usted especifique.

[Además de invocar la función, la acción de AWS Lambdainvocación también convierte cada clave de nivel superior de la carga útil de respuesta recibida de la función Lambda en una variable de salida del flujo de trabajo.](#) A continuación, se puede hacer referencia a estas variables en las siguientes acciones del flujo de trabajo. Si no desea que todas las claves de nivel superior se conviertan en variables, puede usar filtros para especificar las claves exactas. Para obtener más información, consulte la descripción de la [ResponseFilters referencia de acción «AWS Lambda invocar»](#) propiedad en.

Temas

- [Cuándo usar esta acción](#)
- [Ejemplo de flujo de trabajo](#)
- [Añadir la acción «AWS Lambdainvocar»](#)
- [Variables generadas por la acción «AWS Lambdainvocar»](#)
- [definición de acción «AWS Lambdainvocar»](#)

Cuándo usar esta acción

Utilice esta acción si desea añadir una funcionalidad a su flujo de trabajo encapsulada en una función Lambda y ejecutada por ella.

Por ejemplo, es posible que desees que tu flujo de trabajo envíe una `Build started` notificación a un canal de Slack antes de iniciar la creación de tu aplicación. En este caso, tu flujo de trabajo

incluiría una acción de AWS Lambda invocación para invocar una Lambda para enviar la notificación de Slack y una [acción de compilación](#) para crear tu aplicación.

Como otro ejemplo, es posible que desees que tu flujo de trabajo realice un análisis de vulnerabilidades en tu aplicación antes de implementarla. En este caso, utilizaría una acción de compilación para crear la aplicación, una acción de AWS Lambda invocación para invocar una Lambda para buscar vulnerabilidades y una acción de implementación para implementar la aplicación escaneada.

Ejemplo de flujo de trabajo

Note

El siguiente ejemplo de flujo de trabajo tiene únicamente fines ilustrativos y requiere un trabajo de configuración adicional para que funcione correctamente. Su objetivo es dar un ejemplo del aspecto que puede tener un flujo de trabajo cuando se configura con la acción de AWS Lambda invocación.

El siguiente flujo de trabajo incluye la acción de AWS Lambda invocación, junto con una acción de despliegue. El flujo de trabajo envía una notificación de Slack en la que se indica que se ha iniciado una implementación y, a continuación, despliega una aplicación AWS utilizando una plantilla. AWS CloudFormation El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de AWS Lambda invocación (LambdaNotify): al activarse, esta acción invoca la función Notify-Start Lambda en la AWS cuenta y región especificadas (my-aws-account, y). us-west-2 Al invocarse, la función Lambda envía una notificación de Slack que indica que se ha iniciado una implementación.
- Una acción de despliegue de AWS CloudFormation pila (Deploy): al finalizar la acción de AWS Lambda invocación, la acción de despliegue de AWS CloudFormation pila ejecuta la plantilla (cfn-template.yml) para desplegar la pila de aplicaciones. Para obtener más información sobre la acción Implementar AWS CloudFormation una pila, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).

```
Name: codecatalyst-lambda-invoke-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  LambdaNotify:
    Identifier: aws/lambda-invoke@v1
    Environment:
      Name: my-production-environment
    Connections:
      - Name: my-aws-account
        Role: codecatalyst-lambda-invoke-role
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Function: Notify-Start
      AWSRegion: us-west-2

  Deploy:
    Identifier: aws/cfn-deploy@v1
    Environment:
      Name: my-production-environment
    Connections:
      - Name: my-aws-account
        Role: codecatalyst-deploy-role
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      name: my-application-stack
      region: us-west-2
      role-arn: arn:aws:iam::111122223333:role/StackRole
      template: ./cfn-template.yml
      capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
```

Añadir la acción «AWS Lambdainvocar»

Sigue las siguientes instrucciones para añadir la acción de AWS Lambdainvocación a tu flujo de trabajo.

Requisito previo

Antes de empezar, asegúrese de que la AWS Lambda función y el rol de ejecución de Lambda asociado estén listos y disponibles en AWS. Para obtener más información, consulte el tema de la [función de ejecución de Lambda](#) en la Guía AWS Lambda para desarrolladores.

Visual

Para añadir la acción «AWS Lambda invocar» mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de AWS Lambda invocación y realice una de las siguientes acciones:

- Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)

- Elija AWS Lambda invocar. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas, Configuración y Salidas, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [referencia de acción «AWS Lambda invocar»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.

11. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «AWS Lambdainvocar» mediante el editor YAML

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de AWS Lambdainvocación y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

Or (Disyunción)
 - Elija AWS Lambdainvocar. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [referencia de acción «AWS Lambda invocar»](#)
11. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.

12. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Variables generadas por la acción «AWS Lambdainvocar»

Cuando se ejecuta la acción de AWS Lambdainvocación, genera variables que puede utilizar en acciones posteriores del flujo de trabajo. Para obtener más información, consulte [«AWS Lambdainvocar» variables de acción](#) la [Lista de variables predefinidas](#).

definición de acción «AWS Lambdainvocar»

La acción de AWS Lambdainvocación se define como un conjunto de propiedades YAML incluidas en el archivo de definición del flujo de trabajo. Para obtener información sobre estas propiedades, consulte [referencia de acción «AWS Lambda invocar»](#) la [Referencia de definición de flujo de trabajo](#)

Añadir la acción «Renderizar la definición de tareas de Amazon ECS»

En esta sección se describe cómo añadir la acción de definición de tareas Render Amazon ECS a su flujo de trabajo. Esta acción actualiza el campo de imagen de un archivo de [definición de tareas de Amazon Elastic Container Service \(Amazon ECS\)](#) con un nombre de imagen de Docker que proporciona tu flujo de trabajo en tiempo de ejecución.

Note

También puedes usar esta acción para actualizar el `environment` campo de definición de la tarea con variables de entorno.

Temas

- [¿Cuándo usar esta acción](#)
- [Cómo funciona](#)
- [Ejemplo de flujo de trabajo](#)
- [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#)
- [Visualización del archivo de definición de tareas actualizado](#)
- [Variables generadas por la acción «Renderizar la definición de tareas de Amazon ECS»](#)
- [Definición de la acción «Renderizar la definición de tareas de Amazon ECS»](#)

¿Cuándo usar esta acción

Utilízela si tiene un flujo de trabajo que crea y etiqueta una imagen de Docker con contenido dinámico, como un identificador de confirmación o una marca de tiempo.

No utilices esta acción si el archivo de definición de tareas contiene un valor de imagen que siempre es el mismo. En este caso, puede introducir manualmente el nombre de la imagen en el archivo de definición de tareas.

Cómo funciona

Debe utilizar la acción de definición de tareas Renderizar Amazon ECS con las acciones de creación e implementación en Amazon ECS de su flujo de trabajo. Juntas, estas acciones funcionan de la siguiente manera:

1. La acción de creación crea la imagen de Docker y la etiqueta con un nombre, un identificador de confirmación, una marca de tiempo u otro contenido dinámico. Por ejemplo, tu acción de compilación podría tener este aspecto:

```
MyECSWorkflow
  Actions:
    BuildAction:
      Identifier: aws/build@v1
      ...
    Configuration:
      Steps:
        # Build, tag, and push the Docker image...
        - Run: docker build -t MyDockerImage:${WorkflowSource.CommitId} .
        ...
```

En el código anterior, la `docker build -t` directiva indica que hay que crear la imagen de Docker y etiquetarla con el ID de confirmación en el momento de ejecutar la acción. El nombre de la imagen generada podría tener este aspecto:

```
MyDockerImage:a37bd7e
```

2. La acción de definición de tareas Render Amazon ECS añade el nombre de la imagen generada dinámicamente `MyDockerImage:a37bd7e`, al archivo de definición de tareas, de la siguiente manera:

```
{
```



```

    "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
execution-role",
    "containerDefinitions": [
      {
        "name": "codecatalyst-ecs-container",
        "image": MyDockerImage:a37bd7e,
        "essential": true,
        ...
        "portMappings": [
          {
            "hostPort": 80,
            "protocol": "tcp",
            "containerPort": 80
          }
        ]
      }
    ],
    ...
  }

```

Si lo desea, también puede hacer que la acción Render Amazon ECS de definición de tareas añada variables de entorno a la definición de la tarea, de la siguiente manera:

```

{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-
role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": MyDockerImage:a37bd7e,
      ...
      "environment": [
        {
          name": "ECS_LOGLEVEL",
          value": "info"
        }
      ]
    }
  ],
  ...
}

```

Para obtener más información sobre las variables de entorno, consulte [Especificación de variables de entorno](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

3. La acción Implementar en Amazon ECS registra el archivo de definición de tareas actualizado en Amazon ECS. Al registrar el archivo de definición de tareas actualizado, se implementa la nueva imagen MyDockerImage : a37bd7e en Amazon ECS.

Ejemplo de flujo de trabajo

El siguiente es un ejemplo de un flujo de trabajo completo que incluye la acción de definición de tareas Render Amazon ECS, junto con las acciones de creación e implementación. El objetivo del flujo de trabajo es crear e implementar una imagen de Docker en su clúster de Amazon ECS. El flujo de trabajo consta de los siguientes componentes básicos que se ejecutan de forma secuencial:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de compilación (BuildDocker): al activarse, la acción crea la imagen de Docker mediante el Dockerfile, la etiqueta con un ID de confirmación y envía la imagen a Amazon ECR. Para obtener más información sobre la acción de compilación, consulte [Creación mediante flujos de trabajo en CodeCatalyst](#)
- Una acción de definición de tareas de Render Amazon ECS (RenderTaskDef): al finalizar la acción de compilación, esta acción actualiza una existente taskdef.json ubicada en la raíz del repositorio de origen con un valor de image campo que incluye el ID de confirmación correcto. Guarda el archivo actualizado con un nombre de archivo nuevo (task-definition-random-string.json) y, a continuación, crea un artefacto de salida que contiene este archivo. La acción de renderizado también genera una variable llamada task-definition y la establece con el nombre del nuevo archivo de definición de tareas. El artefacto y la variable se utilizarán en la acción de despliegue, que es la siguiente.
- Una acción de implementación en Amazon ECS (**DeployToECS**): al completar la acción de definición de tareas de renderización de Amazon ECS, la acción de implementación en Amazon ECS busca el artefacto de salida generado por la acción de procesamiento (TaskDefArtifact), encuentra el task-definition-random-string.json archivo que contiene y lo registra en el servicio de Amazon ECS. A continuación, el servicio Amazon ECS sigue las instrucciones del task-definition-random-string.json archivo para ejecutar las tareas de Amazon ECS (y los contenedores de imágenes de Docker asociados) dentro del clúster de Amazon ECS.

Name: codecatalyst-ecs-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH

Branches:

- main

Actions:

BuildDocker:

Identifier: aws/build@v1

Environment:

Name: codecatalyst-ecs-environment

Connections:

- Name: codecatalyst-account-connection
- Role: codecatalyst-ecs-build-role

Inputs:

Variables:

- Name: REPOSITORY_URI
- Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-repo
- Name: IMAGE_TAG
- Value: \${WorkflowSource.CommitId}

Configuration:

Steps:

#pre_build:

- Run: echo Logging in to Amazon ECR...
- Run: aws --version
- Run: aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 111122223333.dkr.ecr.us-east-2.amazonaws.com

#build:

- Run: echo Build started on `date`
- Run: echo Building the Docker image...
- Run: docker build -t \$REPOSITORY_URI:latest .
- Run: docker tag \$REPOSITORY_URI:latest \$REPOSITORY_URI:\$IMAGE_TAG

#post_build:

- Run: echo Build completed on `date`
- Run: echo Pushing the Docker images...
- Run: docker push \$REPOSITORY_URI:latest
- Run: docker push \$REPOSITORY_URI:\$IMAGE_TAG

RenderTaskDef:

DependsOn:

- BuildDocker

```
Identifier: aws/ecs-render-task-definition@v1
Inputs:
  Variables:
    - Name: REPOSITORY_URI
      Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
    - Name: IMAGE_TAG
      Value: ${WorkflowSource.CommitId}
Configuration:
  task-definition: taskdef.json
  container-definition-name: codecatalyst-ecs-container
  image: $REPOSITORY_URI:$IMAGE_TAG
# The output artifact contains the updated task definition file.
# The new file is prefixed with 'task-definition'.
# The output variable is set to the name of the updated task definition file.
Outputs:
  Artifacts:
    - Name: TaskDefArtifact
      Files:
        - "task-definition*"
  Variables:
    - task-definition

DeployToECS:
Identifier: aws/ecs-deploy@v1
Environment:
  Name: codecatalyst-ecs-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-ecs-deploy-role
#Input artifact contains the updated task definition file.
Inputs:
  Sources: []
  Artifacts:
    - TaskDefArtifact
Configuration:
  region: us-east-2
  cluster: codecatalyst-ecs-cluster
  service: codecatalyst-ecs-service
  task-definition: ${RenderTaskDef.task-definition}
```

Añadir la acción «Renderizar la definición de tareas de Amazon ECS»

Siga las instrucciones siguientes para añadir la acción de definición de tareas Render Amazon ECS a su flujo de trabajo.

Requisito previo

Antes de empezar, asegúrese de tener un flujo de trabajo que incluya una acción de creación que genere dinámicamente una imagen de Docker. Consulta el [ejemplo anterior de flujo de trabajo](#) para obtener más información.

Visual

Para añadir la acción «Renderizar la definición de tareas de Amazon ECS» mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona Amazon CodeCatalyst.
9. Busque la acción de definición de tareas Render Amazon ECS y realice una de las siguientes acciones:

- Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

- Elija Render la definición de tarea de Amazon ECS. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «Renderizar la definición de tareas de Amazon ECS»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «Renderizar la definición de tareas de Amazon ECS» mediante el editor YAML

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
 2. Elija el proyecto.
 3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 5. Elija Editar.
 6. Elige YAML.
 7. En la parte superior izquierda, seleccione + Acciones para abrir el catálogo de acciones.
 8. En la lista desplegable, seleccione Amazon CodeCatalyst.
 9. Busque la acción de definición de tareas Render Amazon ECS y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
- O bien
- Elija Render la definición de tarea de Amazon ECS. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).

- Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en. [Referencia de acción «Renderizar la definición de tareas de Amazon ECS»](#)
 11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Pasos siguientes

Tras añadir la acción de renderizado, añada la acción Implementar en Amazon ECS a su flujo de trabajo siguiendo las instrucciones que se indican en [Añadir la acción «Implementar en Amazon ECS»](#). Al añadir la acción de despliegue, haga lo siguiente:

1. En la pestaña Entradas de la acción de despliegue, en Artefactos (opcional), seleccione el artefacto que se generó mediante la acción de renderizado. Contiene el archivo de definición de tareas actualizado.

Para obtener más información acerca de los artefactos, consulte [Trabajar con artefactos](#).

2. En la pestaña Configuración de la acción de despliegue, en el campo Definición de la tarea, especifique la siguiente variable de acción: `${action-name}.task-definition` donde *action-name es el nombre* de la acción de renderización, por ejemplo, `RenderTaskDef`. La acción de renderizado establece esta variable en el nuevo nombre del archivo de definición de tareas.

Para obtener más información sobre las variables, consulte [Trabajar con variables](#).

Para obtener más información sobre cómo configurar la acción de despliegue, consulte el [ejemplo de flujo](#) de trabajo anterior.

Visualización del archivo de definición de tareas actualizado

Puede ver el nombre y el contenido del archivo de definición de tareas actualizado.

Para ver el nombre del archivo de definición de tareas actualizado, después de que la acción de definición de tareas Render Amazon ECS lo haya procesado.

1. Busque la ejecución que incluye una acción de renderización completada:
 - a. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
 - b. Elija el proyecto.
 - c. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - d. Elija el nombre del flujo de trabajo que contiene la acción de renderizado. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 - e. Elija una ejecución que incluya la acción de renderización completada.
2. En el diagrama de flujo de trabajo, elija la acción de renderizado.
3. Elija Salidas.
4. Elija variables.
5. Aparece el nombre del archivo de definición de tareas. Tiene un aspecto similar a `task-definition--259-0a2r7gx1TF5X-.json`.

Para ver el contenido del archivo de definición de tareas actualizado

1. Busque la ejecución que incluye una acción de renderización completada:
 - a. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
 - b. Elija el proyecto.
 - c. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - d. Elija el nombre del flujo de trabajo que contiene la acción de renderizado. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 - e. Elija una ejecución que incluya la acción de renderización completada.
2. En la ejecución del flujo de trabajo, en la parte superior, junto a Visual y YAML, selecciona Salidas del flujo de trabajo.
3. En la sección Artefactos, selecciona Descargar junto al artefacto que contiene el archivo de definición de tareas actualizado. Este artefacto tendrá la columna Producido por establecida con el nombre de tu acción de renderizado.

4. Abre el archivo.zip para ver el archivo.json de definición de tareas.

Variables generadas por la acción «Renderizar la definición de tareas de Amazon ECS»

Cuando se ejecuta la acción de definición de tareas Render Amazon ECS, genera variables que puede utilizar en acciones de flujo de trabajo posteriores. Para obtener más información, consulte [Variables de acción «Renderizar la definición de tareas de Amazon ECS»](#) la [Lista de variables predefinidas](#).

Definición de la acción «Renderizar la definición de tareas de Amazon ECS»

La acción de definición de tareas Render Amazon ECS se define como un conjunto de propiedades YAML dentro del archivo de definición de flujo de trabajo. Para obtener información acerca de estas propiedades, consulte [Referencia de acción «Renderizar la definición de tareas de Amazon ECS»](#) la [Referencia de definición de flujo de trabajo](#).

Añadir GitHub acciones

Una GitHub acción se parece mucho a una [CodeCatalyst acción](#), excepto que se desarrolló para su uso con GitHub flujos de trabajo. Para obtener más información sobre GitHub las acciones, consulte la documentación sobre [GitHub las acciones](#).

Puedes usar una GitHub acción junto con CodeCatalyst las acciones nativas en un CodeCatalyst flujo de trabajo.

Hay dos formas de añadir una GitHub acción a un CodeCatalyst flujo de trabajo:

- Puedes seleccionar la GitHub acción de una lista seleccionada en la CodeCatalyst consola. Hay varias GitHub acciones populares disponibles. Para obtener más información, consulte [Añadir una GitHub acción seleccionada](#).
- Si la GitHub acción que quieres usar no está disponible en la CodeCatalyst consola, puedes añadirla mediante una acción de GitHub Acciones.

Una acción de GitHub acciones es una CodeCatalyst acción que envuelve una GitHub acción y la hace compatible con los CodeCatalyst flujos de trabajo.

A continuación se muestra un ejemplo de una acción de GitHub acciones que envuelve la acción de [SuperLinter](#) GitHub:

```
Actions:
```

```
GitHubAction:
  Identifier: aws/github-actions-runner@v1
  Configuration:
    Steps:
      - name: Lint Code Base
        uses: github/super-linter@v4
        env:
          VALIDATE_ALL_CODEBASE: "true"
          DEFAULT_BRANCH: main
```

En el código anterior, la acción CodeCatalyst GitHub Actions (identificada por `aws/github-actions-runner@v1`) envuelve la acción Super-Linter (identificada por `github/super-linter@v4`), lo que hace que funcione en un flujo de trabajo. CodeCatalyst

Para obtener más información, consulte [Añadir la acción «GitHub Acciones»](#).

Todas las GitHub acciones, tanto las seleccionadas como las no seleccionadas, deben estar incluidas en una acción de GitHub Acciones (`aws/github-actions-runner@v1`), como se muestra en el ejemplo anterior. El envoltorio es necesario para que la acción funcione correctamente.

Temas

- [¿En qué se diferencian GitHub las acciones de CodeCatalyst las acciones?](#)
- [¿ GitHub Las acciones pueden interactuar con otras CodeCatalyst acciones del flujo de trabajo?](#)
- [¿Qué GitHub acciones puedo usar?](#)
- [Limitaciones de GitHub las acciones en CodeCatalyst](#)
- [¿Cómo añadir una GitHub acción \(pasos de alto nivel\)?](#)
- [¿Se ejecuta la GitHub acción GitHub?](#)
- [¿También puedo usar GitHub flujos de trabajo?](#)
- [Tutorial: código de Lint mediante una acción GitHub](#)
- [Añadir la acción «GitHub Acciones»](#)
- [Añadir una GitHub acción seleccionada](#)
- [Trabajar con los parámetros de salida de GitHub Actions](#)

¿En qué se diferencian GitHub las acciones de CodeCatalyst las acciones?

GitHub Las acciones que se utilizan dentro de un CodeCatalyst flujo de trabajo no tienen el mismo nivel de acceso e integración ni las mismas CodeCatalyst características (como [los entornos](#) y [los problemas](#)) que CodeCatalyst las acciones. AWS

¿ GitHub Las acciones pueden interactuar con otras CodeCatalyst acciones del flujo de trabajo?

Sí. Por ejemplo, GitHub las acciones pueden usar variables generadas por otras CodeCatalyst acciones como entrada y también pueden compartir los parámetros y artefactos de salida con CodeCatalyst las acciones. Para obtener más información, consulte [Trabajar con los parámetros de salida de GitHub Actions](#).

¿Qué GitHub acciones puedo usar?

Puedes usar cualquier GitHub acción disponible en la CodeCatalyst consola y cualquier GitHub acción disponible en [GitHubMarketplace](#). Si decides usar una GitHub Acción del Marketplace, ten en cuenta las siguientes [limitaciones](#).

Limitaciones de GitHub las acciones en CodeCatalyst

- GitHub Las acciones no se pueden usar con el tipo de [cómputo CodeCatalyst Lambda](#).
- GitHub Las acciones se ejecutan en la imagen de Docker del entorno de ejecución de [noviembre de 2022](#), que incluye herramientas antiguas. Para obtener más información sobre la imagen y las herramientas, consulte. [Trabajar con imágenes de Docker en un entorno de ejecución](#)
- GitHub Las acciones que se basan internamente en el [githubcontexto](#) o que hacen referencia a recursos GitHub específicos no funcionarán en ellas. CodeCatalyst Por ejemplo, las siguientes acciones no funcionarán en CodeCatalyst:
 - Acciones que intentan añadir, cambiar o actualizar GitHub recursos. Algunos ejemplos son las acciones que actualizan las solicitudes de cambios o que crean problemas en ellas GitHub.
 - Casi todas las acciones se muestran en <https://github.com/actions>.
- GitHub Las acciones que sean [acciones de contenedor de Docker](#) funcionarán, pero debe ejecutarlas el usuario predeterminado de Docker (root). No ejecute la acción como usuario 1001. (En el momento de escribir este artículo, el usuario 1001 trabaja en GitHub, pero no en) CodeCatalyst. Para obtener más información, consulta el tema [USER](#) en el [soporte de Dockerfile para GitHub Actions](#).

Para ver una lista de GitHub las acciones disponibles en la CodeCatalyst consola, consulte [Añadir una GitHub acción seleccionada](#)

¿Cómo añado una GitHub acción (pasos de alto nivel)?

Los pasos de alto nivel para añadir una GitHub acción a un CodeCatalyst flujo de trabajo son los siguientes:

1. En el CodeCatalyst proyecto, se crea un flujo de trabajo. En el flujo de trabajo se define cómo crear, probar e implementar la aplicación. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#).
2. En el flujo de trabajo, se añade una GitHub acción seleccionada o se añade la acción GitHub Acciones.
3. Realice una de las siguientes acciones:
 - Si opta por añadir una acción seleccionada, configúrela. Para obtener más información, consulte [Añadir una GitHub acción seleccionada](#).
 - Si eliges añadir una acción no seleccionada, pegas el código YAML de la acción dentro de la GitHub acción GitHub Acciones. Puedes encontrar este código en la página de detalles de la GitHub acción que hayas elegido en [GitHubMarketplace](#). Es probable que tengas que modificar ligeramente el código para que funcione CodeCatalyst. Para obtener más información, consulte [Añadir la acción «GitHub Acciones»](#).
4. (Opcional) Dentro del flujo de trabajo, añades otras acciones, como las acciones de compilación y prueba. Para obtener más información, consulte [Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst](#).
5. El flujo de trabajo se inicia de forma manual o automática mediante un disparador. El flujo de trabajo ejecuta la GitHub acción y cualquier otra acción del flujo de trabajo. Para obtener más información, consulte [Iniciar una ejecución de flujo de trabajo](#).

Para ver los pasos detallados, consulte:

- [Añadir una GitHub acción seleccionada](#).
- [Añadir la acción «GitHub Acciones»](#).

¿Se ejecuta la GitHub acción GitHub?


¿No?. La GitHub acción se ejecuta CodeCatalyst, utilizando [máquinas CodeCatalyst de construcción](#).

¿También puedo usar GitHub flujos de trabajo?

No.

Tutorial: código de Lint mediante una acción GitHub

En este tutorial, añadirás la [GitHub acción Super-Linter](#) a un flujo de trabajo de Amazon CodeCatalyst . La acción Super-Linter inspecciona el código, encuentra áreas en las que el código contiene errores, problemas de formato y construcciones sospechosas y, a continuación, envía los resultados a la consola). CodeCatalyst Tras añadir el linter al flujo de trabajo, ejecute el flujo de trabajo para crear lint en un ejemplo de una aplicación Node.js (). `app.js` A continuación, corrige los problemas detectados y vuelve a ejecutar el flujo de trabajo para comprobar si las soluciones han funcionado.

 Tip

[Considere la posibilidad de usar Super-Linter para bordar archivos YAML, como plantillas. AWS CloudFormation](#)

Temas

- [Requisitos previos](#)
- [Paso 1: Crea un repositorio de código fuente](#)
- [Paso 2: Agregue un archivo `app.js`](#)
- [Paso 3: Crea un flujo de trabajo que ejecute la acción Super-Linter](#)
- [Paso 4: Solucione los problemas encontrados por el Super-Linter](#)
- [Eliminar recursos](#)

Requisitos previos

Antes de empezar, necesitarás:

- Un CodeCatalyst espacio con un conectadoCuenta de AWS. Para obtener más información, consulte [Crear un espacio que admita a los usuarios de AWS Builder ID](#).
- Se llama un proyecto vacío en tu CodeCatalyst espaciocodecatalyst-linter-project. Elige la opción Empezar desde cero para crear este proyecto.

Para obtener más información, consulte [Crear un proyecto vacío en Amazon CodeCatalyst](#).

Paso 1: Crea un repositorio de código fuente

En este paso, se crea un repositorio de origen en CodeCatalyst. Utilizará este repositorio para almacenar el archivo fuente de la aplicación de muestra `app.js`, para este tutorial.

Para obtener más información sobre los repositorios de código fuente, consulte [Crear un repositorio de código fuente](#).

Para crear un repositorio de código fuente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su proyecto, `codecatalyst-linter-project`.
3. En el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.
4. Elija Agregar repositorio y, a continuación, elija Crear repositorio.
5. En Nombre del repositorio, ingresa:

```
codecatalyst-linter-source-repository
```

6. Seleccione Crear.

Paso 2: Agregue un archivo `app.js`

En este paso, añades un `app.js` archivo a tu repositorio de código fuente. `app.js` contiene un código de función que contiene algunos errores que el linter encontrará.

Para añadir el archivo `app.js`

1. En la CodeCatalyst consola, elige tu proyecto, `codecatalyst-linter-project`.
2. En el panel de navegación, selecciona Código y, a continuación, selecciona Repositorios de origen.
3. En la lista de repositorios de origen, selecciona tu repositorio, `codecatalyst-linter-source-repository`

4. En Archivos, selecciona Crear archivo.
5. En el cuadro de texto, introduce el siguiente código:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;
/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 */
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      statusCode: 200,
      'body': JSON.stringify({
        message: 'hello world'
        // location: ret.data.trim()
      })
    }
  } catch (err) {
    console.log(err)
    return err
  }

  return response
}
```

6. En Nombre de archivo, introduzca `app.js`. Mantenga las demás opciones predeterminadas.
7. Elija Commit (Confirmar).

Ahora ha creado un archivo llamado `app.js`.

Paso 3: Crea un flujo de trabajo que ejecute la acción Super-Linter

En este paso, crearás un flujo de trabajo que ejecute la acción Super-Linter al insertar código en tu repositorio de código fuente. El flujo de trabajo consta de los siguientes componentes básicos, que se definen en un archivo YAML:

- Un disparador: este activador inicia la ejecución automática del flujo de trabajo al introducir un cambio en el repositorio de origen. Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).
- Una acción de «GitHub Acciones»: al activarla, la acción GitHub Acciones ejecuta la acción Super-Linter, que a su vez inspecciona todos los archivos del repositorio de origen. Si el filtro detecta un problema, la acción del flujo de trabajo falla.

Para crear un flujo de trabajo que ejecute la acción Super-Linter

1. En la CodeCatalyst consola, elige tu proyecto, `codecatalyst-linter-project`
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Seleccione Crear flujo de trabajo.
4. En Repositorio de origen, elija `codecatalyst-linter-source-repository`.
5. Para Branch, elija `main`.
6. Seleccione Crear.
7. Elimina el código de ejemplo de YAML.
8. Agrega el siguiente YAML:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  SuperLinterAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        github-action-code
```


En el código anterior, `github-action-code` sustitúyalo por el código de acción de SuperLinter, tal y como se indica en los pasos siguientes de este procedimiento.

9. Ve a la [página de Super-Linter](#) en Marketplace. GitHub
10. En `steps`: (minúsculas), busca el código y pégalo en el CodeCatalyst flujo de trabajo (mayúsculas). `Steps`:

Ajusta el código de GitHub acción para que se ajuste a CodeCatalyst los estándares, tal y como se muestra en el código siguiente.

Su CodeCatalyst flujo de trabajo ahora tiene el siguiente aspecto:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  SuperLinterAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Lint Code Base
          uses: github/super-linter@v4
          env:
            VALIDATE_ALL_CODEBASE: "true"
            DEFAULT_BRANCH: main
```

11. (Opcional) Selecciona Validar para asegurarte de que el código YAML es válido antes de confirmarlo.
12. Selecciona Confirmar, introduce un mensaje de confirmación, selecciona tu `codecatalyst-linter-source-repository` repositorio y vuelve a elegir Confirmar.

Ahora ha creado un flujo de trabajo. La ejecución de un flujo de trabajo se inicia automáticamente debido al activador definido en la parte superior del flujo de trabajo.

Para ver la ejecución del flujo de trabajo en curso

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

2. Elija el flujo de trabajo que acaba de crear: `codecatalyst-linter-workflow`
3. En el diagrama de flujo de trabajo, elige SuperLinterAction.
4. Espere a que la acción falle. Se esperaba este error porque Linter encontró problemas en el código.
5. Deje la CodeCatalyst consola abierta y vaya a [Paso 4: Solucione los problemas encontrados por el Super-Linter](#).

Paso 4: Solucione los problemas encontrados por el Super-Linter

El Super-Linter debería haber encontrado problemas en el `app.js` código, así como en el `README.md` archivo incluido en el repositorio fuente.

Para solucionar los problemas encontrados por Linter

1. En la CodeCatalyst consola, seleccione la pestaña Registros y, a continuación, elija Lint Code Base.

Se muestran los registros que generó la acción de Super-Linter.

2. En los registros de Super-Linter, desplácese hacia abajo hasta aproximadamente la línea 90, donde se encuentra el origen de los problemas. Tienen un aspecto similar al siguiente:

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.  
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.  
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body'  
found.  
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but  
found 4.  
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not  
found.
```

3. Corrija `app.js` y `README.md` en su repositorio de código fuente y confirme los cambios.

Tip

Para corregirlo `README.md`, markdown añádelo al bloque de código de la siguiente manera:

```
``markdown  
Setup examples:
```



Los cambios inician otro flujo de trabajo que se ejecuta automáticamente. Espere a que finalice el flujo de trabajo. Si ha solucionado todos los problemas, el flujo de trabajo debería funcionar correctamente.

Eliminar recursos

Limpie CodeCatalyst para eliminar los rastros de este tutorial de su entorno.

Para limpiar en CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Eliminarcodecatalyst-linter-source-repository.
3. Eliminarcodecatalyst-linter-workflow.

En este tutorial, aprendiste a añadir la GitHub acción SuperLinter a un CodeCatalyst flujo de trabajo para borrar parte del código.

Añadir la acción «GitHub Acciones»

Una acción de GitHub acciones es una CodeCatalyst acción que envuelve una GitHub acción y la hace compatible con los CodeCatalyst flujos de trabajo.

Para obtener más información, consulte [Añadir GitHub acciones](#).

Para añadir la acción GitHub Acciones a un flujo de trabajo, sigue estos pasos.

Tip

Para ver un tutorial que muestra cómo utilizar la acción GitHub Acciones, consulte [Tutorial: código de Lint mediante una acción GitHub](#).

Visual

Para añadir la acción «GitHub Acciones» mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona GitHub.
9. Busque la acción GitHub Acciones y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

 - Seleccione GitHub Acciones. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. En las pestañas Entradas y Configuración, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «GitHub Acciones»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de la propiedad de YAML correspondiente) tal como aparece en los editores de YAML y visual.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir la acción «GitHub Acciones» mediante el editor YAML

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona GitHub.
9. Busque la acción GitHub Acciones y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

 - Seleccione GitHub Acciones. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
10. Modifique las propiedades del código YAML según sus necesidades. Encontrará una explicación de cada propiedad disponible en [Referencia de acción «GitHub Acciones»](#)
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Definición de acción «GitHub Acciones»

La acción GitHub Acciones se define como un conjunto de propiedades YAML dentro del archivo de definición de flujo de trabajo. Para obtener información sobre estas propiedades, consulte [Referencia de acción «GitHub Acciones»](#) la [Referencia de definición de flujo de trabajo](#).

Añadir una GitHub acción seleccionada

Una GitHub acción seleccionada es una GitHub acción que está disponible en la CodeCatalyst consola y sirve como ejemplo de cómo utilizar una GitHub acción dentro de un CodeCatalyst flujo de trabajo.

GitHub Las acciones seleccionadas se incluyen en la [acción CodeCatalyst de las GitHub acciones](#) de su autor, que se identifica mediante el `aws/github-actions-runner@v1` identificador. Por ejemplo, este es el aspecto de la GitHub acción seleccionada, [TruffleHog OSS](#):

```
Actions:
  TruffleHogOSS_e8:
    Identifier: aws/github-actions-runner@v1
    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this Workflow as a
source
    Configuration:
      Steps:
        - uses: trufflesecurity/trufflehog@v3.16.0
          with:
            path: ' ' # Required; description: Repository path
            base: ' ' # Required; description: Start scanning from here (usually main
branch).
            head: ' ' # Optional; description: Scan commits until here (usually dev
branch).
            extra_args: ' ' # Optional; description: Extra args to be passed to the
trufflehog cli.
```

En el código anterior, la acción CodeCatalyst GitHub Actions (identificada por `aws/github-actions-runner@v1`) envuelve la acción TruffleHog OSS (identificada por `trufflesecurity/trufflehog@v3.16.0`), lo que hace que funcione en un CodeCatalyst flujo de trabajo.

Para configurar esta acción, sustituya las cadenas vacías que aparecen debajo por `with:` sus propios valores. Por ejemplo:

Actions:**TruffleHogOSS_e8:****Identifier:** aws/github-actions-runner@v1**Inputs:****Sources:**

- WorkflowSource # This specifies that the action requires this Workflow as a source

Configuration:**Steps:**

- uses: trufflesecurity/trufflehog@v3.16.0
 - with:
 - path: ./
 - base: main # Required; description: Start scanning from here (usually main branch).
 - head: HEAD # Optional; description: Scan commits until here (usually dev branch).
 - extra_args: '--debug --only-verified' # Optional; description: Extra args to be passed to the trufflehog cli.

Para añadir una GitHub acción seleccionada a un flujo de trabajo, utilice el siguiente procedimiento. Para obtener información general sobre el uso de GitHub las acciones en un CodeCatalyst flujo de trabajo, consulte [Añadir GitHub acciones](#).

Note

Si tu GitHub acción no aparece en la lista de acciones seleccionadas, puedes añadirla a tu flujo de trabajo mediante la acción GitHub Acciones. Para obtener más información, consulte [Añadir la acción «GitHub Acciones»](#).

Visual

Para añadir una GitHub acción seleccionada mediante el editor visual

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

5. Elija Editar.
 6. Elija Visual.
 7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
 8. En la lista desplegable, selecciona GitHub.
 9. Examine o busque una GitHub acción y realice una de las siguientes acciones:
 - Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

 - Elija el nombre de la GitHub acción. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.
 10. En las pestañas Entradas, Configuración y Salidas, complete los campos según sus necesidades. Para obtener una descripción de cada campo, consulte la [Referencia de acción «GitHub Acciones»](#). Esta referencia proporciona información detallada sobre cada campo (y el valor de propiedad de YAML correspondiente) disponible para la acción GitHubActions, tal como aparece en los editores de YAML y visual.
- Para obtener información sobre las opciones de configuración disponibles para la GitHub acción seleccionada, consulta su documentación.
11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
 12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir una GitHub acción seleccionada mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
8. En la lista desplegable, selecciona GitHub.
9. Examine o busque una GitHub acción y realice una de las siguientes acciones:

- Elija el signo más (+) para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

O bien

- Elija el nombre de la GitHub acción. Aparece el cuadro de diálogo de detalles de la acción. En este cuadro de diálogo:
 - (Opcional) Seleccione Ver código fuente para [ver el código fuente de la acción](#).
 - Seleccione Añadir al flujo de trabajo para añadir la acción al diagrama de flujo de trabajo y abrir su panel de configuración.

10. Modifique las propiedades del código YAML según sus necesidades. En GitHub la. [Referencia de acción «GitHub Acciones»](#)

Para obtener información sobre las opciones de configuración disponibles para la GitHub acción seleccionada, consulta su documentación.

11. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Trabajar con los parámetros de salida de GitHub Actions

Puede integrar [los parámetros GitHub de salida](#) en sus CodeCatalyst flujos de trabajo.

Note

Otra palabra para parámetro de salida es variable. Como GitHub usa el término parámetro de salida en su documentación, también usaremos este término.

Temas

- [Exportación GitHub de un parámetro de salida para que otras acciones puedan usarlo](#)
- [Hacer referencia a un parámetro GitHub de salida](#)

Exportación GitHub de un parámetro de salida para que otras acciones puedan usarlo

Siga las instrucciones siguientes para exportar un parámetro GitHub de salida de modo que esté disponible para su uso en otras acciones del CodeCatalyst flujo de trabajo.

Para exportar un parámetro GitHub de salida

1. Abra un flujo de trabajo y seleccione Editar. Para obtener instrucciones, consulte [Para editar un flujo de trabajo](#).
2. En la acción GitHub Acciones que genera el parámetro de salida que desea exportar, añada una Outputs sección con una Variables propiedad subyacente similar a la siguiente:

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'step-id_output-name'
```

Sustituir:

- *step id* por el valor de la id: propiedad en la steps sección de la GitHub acción.
- *output-name con el nombre* del parámetro de salida. GitHub

Ejemplo

El siguiente ejemplo muestra cómo exportar un parámetro de GitHub salida llamado. SELECTEDCOLOR

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'random-color-generator_SELECTEDCOLOR'
```

Configuration:**Steps:**

- name: Set selected color
run: echo "SELECTEDCOLOR=green" >> \$GITHUB_OUTPUT
id: random-color-generator

Hacer referencia a un parámetro GitHub de salida

Utilice las siguientes instrucciones para hacer referencia a un parámetro GitHub de salida.

Para hacer referencia a un parámetro GitHub de salida

1. Realice los pasos que se indican en [Exportación GitHub de un parámetro de salida para que otras acciones puedan usarlo](#).

El parámetro GitHub de salida ahora está disponible para su uso en otras acciones.

2. Anote el `Variables` valor del parámetro de salida. Incluye un guión bajo (`_`).
3. Consulte el parámetro de salida con la siguiente sintaxis:

```
${action-name.output-name}
```

Sustituir:

- *action-name* por el nombre de la CodeCatalyst GitHub acción que produce el parámetro de salida (no utilices el `name` o `id` de la GitHub acción).
 - *nombre de salida* con el valor del parámetro de salida que anotaste anteriormente.
- `Variables`

Ejemplo

```
BuildActionB:  
  Identifier: aws/build@v1  
  Configuration:  
    Steps:  
      - Run: echo ${MyGitHubAction.random-color-generator_SELECTEDCOLOR}
```

Ejemplo con contexto

El siguiente ejemplo muestra cómo establecer una `SELECTEDCOLOR` variable `GitHubActionA`, generarla y, a continuación, hacer referencia a ella `BuildActionB`.

```
Actions:
  GitHubActionA:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Set selected color
          run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
          id: random-color-generator
    Outputs:
      Variables:
        - 'random-color-generator_SELECTEDCOLOR'

  BuildActionB:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: echo ${GitHubActionA.random-color-generator_SELECTEDCOLOR}
```

Eliminar una acción

Siga las instrucciones siguientes para eliminar una acción de un flujo de trabajo.

Visual

Para eliminar una acción mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, en la acción que desee eliminar, elija el icono de puntos suspensivos verticales y elija Eliminar.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para eliminar una acción mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Busca la sección del YAML que contiene la acción que deseas eliminar.

Selecciona la sección y presiona la tecla de borrar del teclado.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Desarrollo de una acción personalizada

Puede desarrollar una acción personalizada para utilizarla en sus flujos de trabajo mediante el kit de desarrollo de CodeCatalyst acciones (ADK). A continuación, puede publicar la acción en el catálogo de CodeCatalyst acciones para que otros CodeCatalyst usuarios puedan verla y utilizarla en sus flujos de trabajo.

Para desarrollar, probar y publicar una acción (tareas de alto nivel)

1. Instale las herramientas y los paquetes necesarios para desarrollar una acción.
2. Crea un CodeCatalyst repositorio para almacenar tu código de acción.

3. Inicializa la acción. Aquí se indican los archivos fuente necesarios para la acción, incluido un archivo de definición de acciones (`action.yml`) que puede actualizar con su propio código.
4. Arranca el código de acción para obtener las herramientas y bibliotecas necesarias para crear, probar y publicar el proyecto de acción.
5. Crea la acción en tu ordenador local e introduce los cambios en tu CodeCatalyst repositorio.
6. Prueba la acción con pruebas unitarias a nivel local y ejecuta en él el flujo de trabajo generado por ADK. CodeCatalyst
7. Publica la acción en el catálogo de CodeCatalyst acciones pulsando el botón Publicar de la CodeCatalyst consola.

Para ver los pasos detallados, consulta la [Guía para desarrolladores del Amazon CodeCatalyst Action Development Kit](#).

Agrupación de acciones en grupos de acciones

Un grupo de acciones contiene una o más acciones. Agrupar las acciones en grupos de acciones le ayuda a mantener el flujo de trabajo organizado y también le permite configurar las dependencias entre los distintos grupos de acciones.

Note

Los grupos de acciones no se pueden agrupar dentro de otras acciones o grupos de acciones.

Siga las instrucciones siguientes para definir un grupo de acciones.

Visual

No disponible. Selecciona YAML para ver las instrucciones de YAML.

YAML

Para definir un grupo de acciones

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En **Actions**, agrega un código similar al siguiente:

```

Actions:
  action-group-name:
    Actions:
      action-1:
        Identifier: aws/build@v1
        Configuration:
          ...
      action-2:
        Identifier: aws/ecs-deploy@v1
        Configuration:
          ...

```

Para ver otro ejemplo, consulte [Ejemplo: definir dos grupos de acciones](#). Para obtener más información, consulte la descripción [Acciones](#) de la `action-group-name` propiedad en el [Referencia de definición de flujo de trabajo](#).

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Ejemplo: definir dos grupos de acciones

El siguiente ejemplo muestra cómo definir dos grupos de acciones: `BuildAndTest` y `Deploy`. El grupo de `BuildAndTest` acciones incluye dos acciones (`BuildyTest`) y el grupo de `Deploy` acciones también incluye dos acciones (`DeployCloudFormationStackyDeployToECS`).

```

Actions:
  BuildAndTest: # Action group 1
    Actions:
      Build:
        Identifier: aws/build@v1
        Configuration:

```

```
    ...
  Test:
    Identifier: aws/managed-test@v1
    Configuration:
  Deploy: #Action group 2
  Actions:
    DeployCloudFormationStack:
      Identifier: aws/cfn-deploy@v1
      Configuration:
        ...
    DeployToECS:
      Identifier: aws/ecs-deploy@v1
      Configuration:
        ...
```

Configurar acciones para que dependan de otras acciones

De forma predeterminada, al añadir acciones a un flujo de trabajo, se añaden una al lado de la otra en el [editor visual](#). Esto significa que las acciones se ejecutarán en paralelo al iniciar una ejecución de flujo de trabajo. Si desea que las acciones se ejecuten en orden secuencial (y aparezcan verticalmente en el editor visual), debe configurar las dependencias entre ellas. Por ejemplo, puedes configurar una Test acción para que dependa de ella, de modo que la Build acción de prueba se ejecute después de la acción de creación.

Puede configurar las dependencias entre las acciones, entre los grupos de acciones y entre las acciones y los grupos de acciones. También puede configurar one-to-many las dependencias para que una acción dependa de varias otras para poder iniciarse. Consulte la [Directrices para configurar las dependencias](#) para asegurarse de que la configuración de la dependencia se ajusta a la sintaxis YAML del flujo de trabajo.

Temas

- [Configuración de dependencias](#)
- [Directrices para configurar las dependencias](#)
- [Ejemplos](#)

Configuración de dependencias

Siga las instrucciones siguientes para configurar las dependencias entre las acciones de un flujo de trabajo.

Visual

Para configurar las dependencias mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción que dependerá de otra.
8. Seleccione la pestaña Entradas.
9. En Dependencia de: opcional, haga lo siguiente:

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte. [Configurar acciones para que dependan de otras acciones](#)

10. (Opcional) Seleccione Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para configurar las dependencias mediante el editor YAML

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.

6. Elige YAML.
7. En una acción que dependerá de otra, añade un código similar al siguiente:

```
action-name:  
  DependsOn:  
    - action-1
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener pautas generales, consulte [Directrices para configurar las dependencias](#). Para obtener más información, consulte la descripción de la DependsOn propiedad en la sección [Referencia de definición de flujo de trabajo](#) correspondiente a su acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Directrices para configurar las dependencias

Al configurar las dependencias, siga estas pautas:

- Si una acción está dentro de un grupo de acciones, esa acción solo puede depender de otras acciones dentro del mismo grupo de acciones.
- Las acciones y los grupos de acciones pueden depender de otras acciones y grupos de acciones del mismo nivel en la jerarquía de YAML, pero no de un nivel diferente.

Ejemplos

Los siguientes ejemplos muestran cómo configurar las dependencias entre las acciones y los grupos de acciones en el archivo de definición del flujo de trabajo.

Temas

- [Ejemplo: configurar una dependencia simple](#)
- [Ejemplo: configurar un grupo de acciones para que dependa de una acción](#)
- [Ejemplo: configurar un grupo de acciones para que dependa de otro grupo de acciones](#)
- [Ejemplo: configurar un grupo de acciones para que dependa de varias acciones](#)

Ejemplo: configurar una dependencia simple

El siguiente ejemplo muestra cómo configurar una `Test` acción para que dependa de la `Build` acción que utilice la `DependsOn` propiedad.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      ...
  Test:
    DependsOn:
    - Build
    Identifier: aws/managed-test@v1
    Configuration:
      ...
```

Ejemplo: configurar un grupo de acciones para que dependa de una acción

El siguiente ejemplo muestra cómo configurar un grupo de `DeployGroup` acciones para que dependa de la `FirstAction` acción. Observe que la acción y el grupo de acciones están en el mismo nivel.

```
Actions:
  FirstAction: #An action outside an action group
    Identifier: aws/github-actions-runner@v1
    Configuration:
      ...
  DeployGroup: #An action group containing two actions
    DependsOn:
    - FirstAction
    Actions:
      DeployAction1:
        ...
      DeployAction2:
        ...
```

Ejemplo: configurar un grupo de acciones para que dependa de otro grupo de acciones

El siguiente ejemplo muestra cómo configurar un grupo de `DeployGroup` acciones para que dependa del grupo de `BuildAndTestGroup` acciones. Observe que los grupos de acciones están en el mismo nivel.

```

Actions:
  BuildAndTestGroup: # Action group 1
    Actions:
      BuildAction:
      ...
      TestAction:
      ...
  DeployGroup: #Action group 2
    DependsOn:
      - BuildAndTestGroup
    Actions:
      DeployAction1:
      ...
      DeployAction2:
      ...

```

Ejemplo: configurar un grupo de acciones para que dependa de varias acciones

El siguiente ejemplo muestra cómo configurar un grupo de DeployGroup acciones para que dependa de la FirstAction acción, de la SecondAction acción y del grupo de BuildAndTestGroup acciones. Observe que DeployGroup está al mismo nivel que FirstActionSecondAction, yBuildAndTestGroup.

```

Actions:
  FirstAction: #An action outside an action group
  ...
  SecondAction: #Another action
  ...
  BuildAndTestGroup: #Action group 1
    Actions:
      Build:
      ...
      Test:
      ...
  DeployGroup: #Action group 2
    DependsOn:
      - FirstAction
      - SecondAction
      - BuildAndTestGroup
    Actions:
      DeployAction1:
      ...

```

```
DeployAction2:  
...
```

Visualización del código fuente de una acción

Puedes ver el código fuente de una acción para asegurarte de que no contiene código peligroso, vulnerabilidades de seguridad u otros defectos.

Siga las instrucciones siguientes para ver el código fuente de una acción [CodeCatalyst](#), de [CodeCatalyst Labs](#) o [de terceros](#).

Note

Para ver el código fuente de una [GitHubacción](#), ve a la página de la acción en [GitHub Marketplace](#). La página incluye un enlace al repositorio de la acción, donde puedes encontrar el código fuente de la acción.

Note

No puedes ver el código fuente de las siguientes CodeCatalyst acciones: [compilar](#), [probar](#) o [GitHub Actions](#).

Note

AWSno respalda ni garantiza el código de acción de GitHub Actions ni de las acciones de terceros.

Para ver el código fuente de una acción

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. Busca la acción cuyo código deseas ver:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija el nombre de cualquier flujo de trabajo o cree uno. Para obtener información sobre la creación de un flujo de trabajo, consulte [Crear, editar y eliminar un flujo de trabajo](#).

- c. Elija Editar.
- d. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
- e. En la lista desplegable, selecciona Amazon CodeCatalyst para ver CodeCatalyst las acciones de CodeCatalyst Labs y de terceros.
- f. Busca una acción y elige su nombre. No elija el signo más (+).

Aparecen los detalles de la acción.

4. En el cuadro de diálogo de detalles de la acción, cerca de la parte inferior, selecciona Descargar.

Aparece una página en la que se muestra el bucket de Amazon S3 en el que reside el código fuente de la acción. Para obtener información sobre Amazon S3, consulte [¿Qué es Amazon S3?](#) en la Guía del usuario de Amazon Simple Storage Service.

5. Inspeccione el código para asegurarse de que cumple sus expectativas de calidad y seguridad.

Trabajar con versiones de acción

De forma predeterminada, cuando añades una acción a un flujo de trabajo, Amazon CodeCatalyst añade la versión completa al archivo de definición del flujo de trabajo con el siguiente formato:

vmajor.minor.patch

Por ejemplo:

```
My-Build-Action:  
  Identifier: aws/build@v1.0.0
```

Puede acortar la versión completa de la `Identifier` propiedad para que el flujo de trabajo utilice siempre la última versión secundaria o de parche de la acción.

Por ejemplo, si especifica:

```
My-CloudFormation-Action:  
  Identifier: aws/cfn-deploy@v1.0
```

... y la última versión del parche es `1.0.4` la que se utilizará para la acción `1.0.4`. Si se lanza una versión posterior, por ejemplo `1.0.5`, la acción se usará `1.0.5`. Si se lanza una versión secundaria, por ejemplo `1.1.0`, la acción seguirá utilizándose `1.0.5`.

Para obtener instrucciones detalladas sobre cómo especificar las versiones, consulte uno de los temas siguientes.

Temas

- [Especificar la versión de la acción que se va a utilizar](#)
- [Determinar qué versiones de una acción están disponibles](#)

Especificar la versión de la acción que se va a utilizar

Siga las instrucciones siguientes para indicar qué versión de una acción desea que utilice su flujo de trabajo. Puede especificar la última versión principal o secundaria, o una versión de parche específica.

Recomendamos utilizar la última versión secundaria o de parche de una acción.

Visual


No disponible. Selecciona YAML para ver las instrucciones de YAML.

YAML


Para configurar un flujo de trabajo para usar la última versión de una acción o una versión de parche específica

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Busca la acción cuya versión quieres editar.
8. Busca la `Identifier` propiedad de la acción y establece la versión en una de las siguientes opciones:
 - `action-identifier @v major`: utilice esta sintaxis para que el flujo de trabajo utilice una versión principal específica y permita elegir automáticamente las versiones secundarias y de parche más recientes.

- *identificador de acción @v major.minor*: utilice esta sintaxis para que el flujo de trabajo utilice una versión secundaria específica y permita elegir automáticamente la última versión del parche.
- *identificador de acción @v major.minor.parche*: utilice esta sintaxis para que el flujo de trabajo utilice una versión de parche específica.

 Note

Si no está seguro de qué versiones están disponibles, consulte [Determinar qué versiones de una acción están disponibles](#).

 Note

No puede omitir la versión principal.

9. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Determinar qué versiones de una acción están disponibles

Siga las instrucciones siguientes para determinar qué versiones de una acción están disponibles para su uso en un flujo de trabajo.

Visual

Para determinar qué versiones de acción están disponibles

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. Busca la acción cuyas versiones deseas ver:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija el nombre de cualquier flujo de trabajo o cree uno. Para obtener información sobre la creación de un flujo de trabajo, consulte [Crear, editar y eliminar un flujo de trabajo](#).

- c. Elija Editar.
- d. En la parte superior izquierda, selecciona + Acciones para abrir el catálogo de acciones.
- e. En la lista desplegable, selecciona Amazon CodeCatalyst para ver CodeCatalyst, CodeCatalyst Labs y acciones de terceros, o elige GitHubver GitHub acciones seleccionadas.
- f. Busca una acción y elige su nombre. No elija el signo más (+).

Aparecen los detalles de la acción.

4. En el cuadro de diálogo de detalles de la acción, situado en la parte superior derecha, seleccione la lista desplegable Versiones para ver una lista de las versiones disponibles de la acción.

YAML

No disponible. Seleccione «visual» para ver las instrucciones del editor visual.

Trabajar con artefactos

Un artefacto es el resultado de una acción de flujo de trabajo y, por lo general, consiste en una carpeta o un archivo de archivos. Los artefactos son importantes porque permiten compartir archivos e información entre acciones.

Por ejemplo, puede que tengas una acción de creación que genere un `sam-template.yml` archivo, pero que desees una acción de despliegue para usarlo. En este escenario, utilizaría un artefacto para permitir que la acción de creación comparta el `sam-template.yml` archivo con la acción de despliegue. El código podría tener un aspecto similar al siguiente:

```
Actions:
  BuildAction:
    Identifier: aws/build@v1
    Steps:
      - Run: sam package --output-template-file sam-template.yml
    Outputs:
      Artifacts:
        - Name: MYARTIFACT
          Files:
            - sam-template.yml
  DeployAction:
```

```
Identifier: aws/cfn-deploy@v1
Inputs:
Artifacts:
  - MYARTIFACT
Configuration:
  template: sam-template.yml
```

En el código anterior, la acción build (BuildAction) genera un `sam-template.yml` archivo y, a continuación, lo añade a un artefacto de salida denominado MYARTIFACT. Una acción de despliegue posterior (DeployAction) se especifica MYARTIFACT como entrada, lo que le da acceso al `sam-template.yml` archivo.

Temas

- [¿Puedo compartir artefactos sin especificarlos como salidas y entradas?](#)
- [¿Puedo compartir artefactos entre flujos de trabajo?](#)
- [Definir un artefacto de salida](#)
- [Definir un artefacto de entrada](#)
- [Hacer referencia a los archivos de un artefacto](#)
- [¿Descargando artefactos](#)
- [Ejemplos](#)

¿Puedo compartir artefactos sin especificarlos como salidas y entradas?

Sí, puedes compartir artefactos entre acciones sin especificarlos en las Inputs secciones Outputs y del código YAML de tus acciones. Para ello, debes activar el uso compartido del ordenador. Para obtener más información sobre el uso compartido de equipos y cómo especificar los artefactos cuando está activado, consulte [Compartir el cómputo entre las acciones](#).

Note

Si bien la función de cómputo compartido te permite simplificar el código YAML del flujo de trabajo al eliminar la necesidad de Inputs usar las secciones Outputs y, esta función tiene limitaciones que debes tener en cuenta antes de activarla. Para obtener información sobre estas limitaciones, consulte [Consideraciones sobre el uso compartido de la computación](#).

¿Puedo compartir artefactos entre flujos de trabajo?

No, no puede compartir artefactos entre diferentes flujos de trabajo; sin embargo, puede compartir artefactos entre acciones dentro del mismo flujo de trabajo.

Definir un artefacto de salida

Siga las instrucciones siguientes para definir el artefacto que desee que genere una acción. A continuación, este artefacto estará disponible para que lo utilicen otras acciones.

Note

No todas las acciones admiten artefactos de salida. Para determinar si su acción los admite, siga las instrucciones del editor visual que aparecen a continuación y compruebe si la acción incluye un botón de artefactos de salida en la pestaña Salidas. En caso afirmativo, se admiten los artefactos de salida.

Visual

Para definir un artefacto de salida mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción que producirá el artefacto.
8. Elija la pestaña Salidas.
9. En Artefactos, selecciona Añadir artefacto.
10. Selecciona Añadir artefacto e introduce la información en los campos de la siguiente manera.

Construye el nombre del artefacto

Especifica el nombre de un artefacto generado por la acción. Los nombres de los artefactos deben ser únicos dentro de un flujo de trabajo y están limitados a caracteres alfanuméricos

(a-z, A-Z, 0-9) y guiones bajos (_). No se permiten espacios, guiones (-) ni otros caracteres especiales. No puede utilizar comillas para activar los espacios, guiones y otros caracteres especiales en los nombres de los artefactos de salida.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#)

Archivos producidos por build

Especifique los archivos que se CodeCatalyst incluyen en el artefacto que genera la acción. Estos archivos los genera la acción del flujo de trabajo cuando se ejecuta y también están disponibles en el repositorio de origen. Las rutas de los archivos pueden residir en un repositorio de origen o en un artefacto de una acción anterior y son relativas al repositorio de origen o a la raíz del artefacto. Puede utilizar patrones globales para especificar las rutas. Ejemplos:

- Para especificar un único archivo que esté en la raíz de la ubicación de compilación o de la ubicación del repositorio de origen, utilice `my-file.jar`.
- Para especificar un único archivo en un subdirectorio, use `directory/my-file.jar` o `directory/subdirectory/my-file.jar`.
- Para especificar todos los archivos, utilice `**/*`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.
- Para especificar todos los archivos y directorios de un directorio denominado `directory`, utilice `directory/**/*`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.
- Para especificar todos los archivos de un directorio denominado `directory`, pero no ninguno de sus subdirectorios, utilice `directory/*`.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otro carácter especial, escriba la ruta entre comillas dobles (). "" Para obtener más información sobre los caracteres especiales, consulte [Pautas y convenciones de sintaxis](#)

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Note

Puede que tenga que añadir un prefijo a la ruta del archivo para indicar en qué artefacto o fuente puede encontrarlo. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

11. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para definir un artefacto de salida mediante el editor YAML

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una acción de flujo de trabajo, añada un código similar al siguiente:

```
action-name:
  Outputs:
    Artifacts:
      - Name: artifact-name
        Files:
          - file-path-1
          - file-path-2
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener más información, consulta la [Referencia de definición de flujo de trabajo](#) sección para tu acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Definir un artefacto de entrada

Si desea utilizar un artefacto generado por otra acción, debe especificarlo como entrada para la acción actual. Es posible que puedas especificar varios artefactos como entrada; depende de la acción. Para obtener más información, consulta la sección [Referencia de definición de flujo de trabajo](#) Para tu acción.

Note

No puede hacer referencia a artefactos de otros flujos de trabajo.

Siga las instrucciones siguientes para especificar un artefacto de otra acción como entrada para la acción actual.

Requisito previo

Antes de empezar, asegúrate de que has generado el artefacto de la otra acción. Para obtener más información, consulte [Definir un artefacto de salida](#). Al generar el artefacto, estará disponible para que lo usen otras acciones.

Visual

Para especificar un artefacto como entrada a una acción (editor visual)


1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.

7. En el diagrama de flujo de trabajo, elija la acción en la que desee especificar un artefacto como entrada.
8. Elija Entradas.
9. En Artefactos (opcional), haga lo siguiente:

Especifique los artefactos de acciones anteriores que desee proporcionar como entrada a esta acción. Estos artefactos ya deben estar definidos como artefactos de salida en acciones anteriores.

Si no especifica ningún artefacto de entrada, debe especificar al menos un repositorio de origen en `!action-name/Inputs/Sources`.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

 Note

Si la lista desplegable Artefactos: opcional no está disponible (editor visual) o si se producen errores al validar el YAML (editor de YAML), es posible que la acción solo admita una entrada. En este caso, intenta eliminar la entrada de origen.

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para especificar un artefacto como entrada a una acción (editor YAML)

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.

7. En la acción en la que desee especificar el artefacto como entrada, añada un código similar al siguiente:

```
action-name:
  Inputs:
    Artifacts:
      - artifact-name
```

Para obtener más ejemplos, consulte [Ejemplos](#).

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Hacer referencia a los archivos de un artefacto

Si tiene un archivo que reside dentro de un artefacto y necesita hacer referencia a este archivo en una de sus acciones de flujo de trabajo, complete el siguiente procedimiento.

Note

Véase también [Hacer referencia a archivos de un repositorio fuente](#).

Para hacer referencia a los archivos de un artefacto

- En la acción en la que desee hacer referencia a un archivo, añada un código similar al siguiente:

```
Actions:
  My-action:
    Inputs:
      Sources:
        - WorkflowSource
      Artifacts:
        - artifact-name
    Configuration:
      Steps:
        - run: cd $CATALYST_SOURCE_DIR_artifact-name/build-output && cat file.txt
```


En el código anterior, la acción busca el archivo en el build-output directorio del *artefact-name* y lo muestra. `file.txt`

Para obtener más ejemplos, consulte [Ejemplos](#).

Note

Es posible que puedas omitir el `$CATALYST_SOURCE_DIR_`*artefact-name*/ prefijo en función de cómo hayas configurado la acción. Para obtener más información, consulta la siguiente guía.

Guía sobre cómo hacer referencia a las variables:

- Si tu acción incluye solo un elemento `Inputs` (por ejemplo, incluye un artefacto de entrada y ningún origen), puedes omitir el prefijo y especificar solo la ruta del archivo en relación con la raíz del artefacto.
- También puedes omitir el prefijo si el archivo reside en la entrada principal. La entrada principal es el artefacto de entrada o el `WorkflowSource` primer artefacto de entrada de la lista, si no lo hay. `WorkflowSource`
- El prefijo puede variar en función de la acción que utilices. Para obtener más información, consulte la siguiente tabla.

Tipo de acción	Prefijo de ruta de archivo que se va a utilizar	Ejemplo
Acción de construcción , acción de prueba	<code>\$CATALYST_SOURCE_DIR_</code> <code>IR_ </code> <i>artefact-name</i> /	<code>\$CATALYST_SOURCE_DIR_</code> <code>IR_MyArtifact/</code> folder1/ <code>file.txt</code>
Todas las demás acciones	<code>\$CATALYST_SOURCE_DIR_</code> <code>IR_ </code> <i>artefact-name</i> / o <code>/artifacts/ </code> <i>current-action-name</i> / <code>artefact-</code>	<code>\$CATALYST_SOURCE_DIR_</code> <code>IR_MyArtifact/</code> folder1/ <code>file.txt</code> o

Tipo de acción	Prefijo de ruta de archivo que se va a utilizar	Ejemplo
	<i>name / (esta ruta es un enlace simbólico a \$CATALYST_SOURCE_DIR_ artifact-name/)</i>	/artifacts/MyCurrentAction/MyArtifact/folder1/file.txt

¿Descargando artefactos

Puede descargar e inspeccionar los artefactos generados por las acciones de su flujo de trabajo para solucionar problemas. Hay dos tipos de artefactos que puedes descargar:

- Artefactos de origen: un artefacto que contiene una instantánea del contenido del repositorio de origen tal como estaba cuando se inició la ejecución.
- Artefactos del flujo de trabajo: un artefacto definido en la Outputs propiedad del archivo de configuración del flujo de trabajo.

Para descargar los artefactos generados por el flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. A En el nombre del flujo de trabajo, seleccione Ejecuciones.
6. En Historial de ejecuciones, en la columna ID de ejecución, elija una ejecución. Por ejemplo, Run-95a4d.
7. En el nombre de la carrera, selecciona Artifacts.
8. Junto a un artefacto, selecciona Descargar. Se descarga un archivo comprimido. Su nombre de archivo consta de siete caracteres aleatorios.
9. Extraiga el archivo con la utilidad de extracción de archivos que elija.

Ejemplos

Los siguientes ejemplos muestran cómo generar y hacer referencia a los artefactos en el archivo de definición del flujo de trabajo.

Temas

- [Ejemplo: generar un artefacto](#)
- [Ejemplo: introducir un artefacto generado por otra acción](#)
- [Ejemplo: hacer referencia a archivos en varios artefactos](#)
- [Ejemplo: hacer referencia a un archivo en un único artefacto](#)
- [Ejemplo: hacer referencia a un archivo en un artefacto cuando hay un WorkflowSource](#)

Ejemplo: generar un artefacto

El siguiente ejemplo muestra cómo generar un artefacto que incluye dos archivos.jar.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT1
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
```

Ejemplo: introducir un artefacto generado por otra acción

En el siguiente ejemplo, se muestra cómo generar un artefacto al que se ha llamado y cómo BuildActionA se introduce ARTIFACT4 en él. BuildActionB

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT4
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
```

```
BuildActionB:
  Identifier: aws/build@v1
  Inputs:
    Artifacts:
      - ARTIFACT4
  Configuration:
```

Ejemplo: hacer referencia a archivos en varios artefactos

El siguiente ejemplo muestra cómo generar dos artefactos denominados ART5 y ART6 en y, a continuación BuildActionC, hacer referencia a dos archivos denominados file5.txt (en artefactoART5) y file6.txt (en artefactoART6) in BuildActionD (underSteps).

Note

Para obtener más información sobre cómo hacer referencia a archivos, consulte [Hacer referencia a los archivos de un artefacto](#)

Note

Aunque en el ejemplo se muestra el \$CATALYST_SOURCE_DIR_ART5 prefijo que se está utilizando, puede omitirlo. Esto se debe a que ART5 es la entrada principal. Para obtener más información sobre la entrada principal, consulte [Hacer referencia a los archivos de un artefacto](#).

```
Actions:
  BuildActionC:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ART5
          Files:
            - build-output/file5.txt
        - Name: ART6
          Files:
            - build-output/file6.txt
  BuildActionD:
    Identifier: aws/build@v1
    Inputs:
```

```

Artifacts:
  - ART5
  - ART6
Configuration:
Steps:
  - run: cd $CATALYST_SOURCE_DIR_ART5/build-output && cat file5.txt
  - run: cd $CATALYST_SOURCE_DIR_ART6/build-output && cat file6.txt

```

Ejemplo: hacer referencia a un archivo en un único artefacto

En el siguiente ejemplo, se muestra cómo generar un artefacto denominado ART7 in yBuildActionE, a continuación, hacer referencia file7.txt (en el artefactoART7) en BuildActionF (under). Steps

Observe que la referencia no requiere el prefijo del \$CATALYST_SOURCE_DIR_ *nombre del artefacto* delante del directorio, como lo hacía antes. build-output [Ejemplo: hacer referencia a archivos en varios artefactos](#) Esto se debe a que solo hay un elemento especificado en. Inputs

Note

Para obtener más información sobre cómo hacer referencia a archivos, consulte [Hacer referencia a los archivos de un artefacto](#).

```

Actions:
BuildActionE:
  Identifier: aws/build@v1
  Outputs:
    Artifacts:
      - Name: ART7
      Files:
        - build-output/file7.txt
BuildActionF:
  Identifier: aws/build@v1
  Inputs:
    Artifacts:
      - ART7
  Configuration:
    Steps:
      - run: cd build-output && cat file7.txt

```

Ejemplo: hacer referencia a un archivo en un artefacto cuando hay un WorkflowSource

En el siguiente ejemplo, se muestra cómo generar un artefacto denominado ART8 in yBuildActionG, a continuación, hacer referencia file8.txt (en el artefactoART8) en BuildActionH (under). Steps

Observe cómo la referencia requiere el prefijo del \$CATALYST_SOURCE_DIR_ *nombre del artefacto, como en el* caso anterior. [Ejemplo: hacer referencia a archivos en varios artefactos](#) Esto se debe a que hay varios elementos especificados en Inputs (una fuente y un artefacto), por lo que necesitará el prefijo para indicar dónde buscar el archivo.

Note

Para obtener más información sobre cómo hacer referencia a archivos, consulte. [Hacer referencia a los archivos de un artefacto](#)

Actions:

BuildActionG:

Identifier: aws/build@v1

Outputs:

Artifacts:

- Name: ART8

Files:

- build-output/file8.txt

BuildActionH:

Identifier: aws/build@v1

Inputs:

Sources:

- WorkflowSource

Artifacts:

- ART8

Configuration:

Steps:

- run: cd \$CATALYST_SOURCE_DIR_ART8/build-output && cat file8.txt

Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución

En un CodeCatalyst flujo de trabajo, puede especificar la imagen del entorno de procesamiento y tiempo de ejecución que se CodeCatalyst utiliza para ejecutar las acciones del flujo de trabajo.

El término cómputo se refiere al motor informático (la CPU, la memoria y el sistema operativo) que administra y mantiene CodeCatalyst para ejecutar las acciones del flujo de trabajo.

Una imagen de entorno de ejecución es un contenedor de Docker en el que CodeCatalyst se ejecutan las acciones del flujo de trabajo. El contenedor Docker se ejecuta sobre la plataforma de procesamiento elegida e incluye un sistema operativo y herramientas adicionales que una acción de flujo de trabajo podría necesitar, como Node.js y tar. AWS CLI

Temas

- [Trabajar con computación](#)
- [Compartir el cómputo entre las acciones](#)
- [Trabajar con imágenes de Docker en un entorno de ejecución](#)

Trabajar con computación

La computación se refiere al motor informático (la CPU, la memoria y el sistema operativo) que administra y mantiene CodeCatalyst para ejecutar las acciones del flujo de trabajo.

Note

Si el procesamiento se define como una propiedad del flujo de trabajo, no se puede definir como una propiedad de ninguna acción de ese flujo de trabajo. Del mismo modo, si el cálculo se define como una propiedad de cualquier acción, no se puede definir en el flujo de trabajo.

Temas

- [Acerca de los tipos de procesamiento](#)
- [Acerca de las flotas informáticas](#)
- [Propiedades de la flota bajo demanda](#)
- [Propiedades de la flota aprovisionada](#)

- [Crear, editar y eliminar una flota aprovisionada](#)
- [Asignar una flota aprovisionada o un procesamiento bajo demanda a una acción](#)

Acerca de los tipos de procesamiento

CodeCatalyst ofrece los siguientes tipos de procesamiento:

- Amazon EC2
- AWS Lambda

Amazon EC2 ofrece una flexibilidad optimizada durante las ejecuciones de acción y Lambda ofrece velocidades optimizadas de inicio de las acciones. Lambda permite ejecutar acciones de flujo de trabajo más rápidas debido a una latencia de inicio más baja. Lambda le permite ejecutar flujos de trabajo básicos que pueden crear, probar e implementar aplicaciones sin servidor con tiempos de ejecución comunes. Estos tiempos de ejecución incluyen Node.js, Python, Java, .NET y Go. Sin embargo, hay algunos casos de uso que Lambda no admite y, si le afectan, utilice el tipo de procesamiento Amazon EC2:

- Lambda no admite imágenes del entorno de ejecución de un registro específico.
- Lambda no admite herramientas que requieran permisos de root. Para herramientas como yum o rpm, utilice el tipo de procesamiento Amazon EC2 u otras herramientas que no requieran permisos de root.
- Lambda no admite compilaciones ni ejecuciones de Docker. No se admiten las siguientes acciones que utilizan imágenes de Docker: Deploy AWS CloudFormation stack, Deploy to Amazon ECS, Amazon S3 publish, AWS CDK bootstrap, AWS CDK deploy, AWS Lambda invoke y Actions. GitHub Actions Las acciones basadas en Docker que se ejecutan dentro de CodeCatalyst GitHub Actions Action tampoco son compatibles con la computación Lambda. Puede utilizar alternativas que no requieran permisos de root, como Podman.
- Lambda no admite la escritura en archivos externos. /tmp Al configurar las acciones del flujo de trabajo, puede volver a configurar las herramientas para instalarlas o escribirlas. /tmp Si tiene una acción de compilación que se instala con npm, asegúrese de configurarla para que se instale en. /tmp
- Lambda no admite tiempos de ejecución superiores a 15 minutos.

Acerca de las flotas informáticas

CodeCatalyst ofrece las siguientes flotas de cómputo:

- Flotas bajo demanda
- Flotas aprovisionadas

En el caso de las flotas bajo demanda, cuando se inicia una acción de flujo de trabajo, el flujo de trabajo aprovisiona los recursos que necesita. Las máquinas se destruyen cuando finaliza la acción. Solo pagas por el número de minutos que llevas a cabo tus acciones. Las flotas bajo demanda están completamente administradas e incluyen capacidades de escalado automático para gestionar los picos de demanda.

CodeCatalyst también ofrece flotas aprovisionadas que contienen máquinas alimentadas por Amazon EC2 que son mantenidas por CodeCatalyst. Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Con las flotas aprovisionadas, sus máquinas están siempre en funcionamiento e incurrirán en costes mientras estén aprovisionadas.

Para crear, actualizar o eliminar una flota, debe tener el rol de administrador de espacios o el rol de administrador de proyectos.

Propiedades de la flota bajo demanda

CodeCatalyst proporciona las siguientes flotas bajo demanda:

Nombre	Sistema operativo	Arquitectura	vCPU	Memoria (GiB)	Espacio en disco	Tipos de cómputo compatibles
Linux.Arm64.Large	Amazon Linux 2	Arm64	2	4	64 GB	Amazon EC2
					10 GB	Lambda
Linux.Arm64.XLarge	Amazon Linux 2	Arm64	4	8	128 GB	Amazon EC2
					10 GB	Lambda

Nombre	Sistema operativo	Arquitectura	vCPU	Memoria (GiB)	Espacio en disco	Tipos de cómputo compatibles
Linux.Arm64.2XLarge	Amazon Linux 2	Arm64	8	16	128 GB	Amazon EC2
Linux.x86-64.Large	Amazon Linux 2	x86-64	2	4	64 GB	Amazon EC2
					10 GB	Lambda
Linux.x86-64.XLarge	Amazon Linux 2	x86-64	4	8	128 GB	Amazon EC2
					10 GB	Lambda
Linux.x86-64.2XLarge	Amazon Linux 2	x86-64	8	16	128 GB	Amazon EC2

Si no se selecciona ninguna flota, CodeCatalyst utiliza `Linux.x86-64.Large`.

Propiedades de la flota aprovisionada

Una flota aprovisionada contiene las siguientes propiedades:

Sistema operativo

El sistema operativo. A continuación, se muestran los sistemas operativos disponibles:

- Amazon Linux 2
- Windows Server 2022

Note

Las flotas de Windows solo se admiten en la acción de construcción. Actualmente, otras acciones no son compatibles con Windows.

Arquitectura

Arquitectura del procesador. Están disponibles las siguientes arquitecturas:

- x86_64
- Arm64

Tipo de máquina

El tipo de máquina de cada instancia. Dispone de los siguientes tipos de máquina:

vCPU	Memoria (GiB)	Espacio en disco	Sistema operativo
2	4	64 GB	Amazon Linux 2
4	8	128 GB	Amazon Linux 2
			Windows Server 2022
8	16	128 GB	Amazon Linux 2
			Windows Server 2022

Capacidad

El número inicial de máquinas asignadas a la flota, que define el número de acciones que se pueden ejecutar en paralelo.

Modo de escalado

Define el comportamiento cuando el número de acciones supera la capacidad de la flota.

Aprovisionamiento de capacidad adicional bajo demanda

Se configuran máquinas adicionales a pedido, que se amplían automáticamente en respuesta a las nuevas acciones que se estén ejecutando y, a continuación, se reducen a la capacidad base a medida que finalizan las acciones. Esto puede conllevar costos adicionales, ya que se paga por minuto por cada máquina en ejecución.

Esperar hasta que se disponga de capacidad de flota adicional

Las ejecuciones de acciones se colocan en una cola hasta que haya una máquina disponible. Esto limita los costos adicionales, ya que no se asignan máquinas adicionales.

Crear, editar y eliminar una flota aprovisionada

Siga las siguientes instrucciones para crear, editar y eliminar flotas aprovisionadas.

Note

Las flotas aprovisionadas se desactivarán tras 2 semanas de inactividad. Si se vuelven a utilizar, se reactivarán automáticamente, pero esta reactivación puede provocar una latencia.

Para crear una flota aprovisionada

1. En el panel de navegación, elija CI/CD y, a continuación, elija Compute.
2. Elija Crear flota aprovisionada.
3. En el campo de texto del nombre de la flota aprovisionada, introduzca un nombre para la flota.
4. En el menú desplegable Sistema operativo, elija el sistema operativo.
5. En el menú desplegable Tipo de máquina, elija el tipo de máquina para su máquina.
6. En el campo de texto Capacidad, introduzca el número máximo de máquinas de la flota.
7. En el menú desplegable Modos de escalado, elija el comportamiento de desbordamiento deseado. Para obtener más información acerca de estos campos, consulte [Propiedades de la flota aprovisionada](#).
8. Seleccione Crear.


Tras crear la flota aprovisionada, está listo para asignarla a una acción. Para obtener más información, consulte [Asignar una flota aprovisionada o un procesamiento bajo demanda a una acción](#).

Para editar una flota aprovisionada

1. En el panel de navegación, elija CI/CD y, a continuación, elija Compute.
2. En la lista de flotas aprovisionadas, elija la flota que desee editar.

3. Elija Editar.
4. En el campo de texto Capacidad, introduzca el número máximo de máquinas de la flota.
5. En el menú desplegable Modos de escalado, elija el comportamiento de desbordamiento deseado. Para obtener más información acerca de estos campos, consulte [Propiedades de la flota aprovisionada](#).
6. Seleccione Guardar.

Para eliminar una flota aprovisionada

 Warning

Antes de eliminar una flota aprovisionada, elimínala de todas las acciones eliminando la `Fleet` propiedad del código YAML de la acción. Cualquier acción que siga haciendo referencia a una flota aprovisionada después de eliminarla fallará la próxima vez que se ejecute la acción.

1. En el panel de navegación, elija CI/CD y, a continuación, elija Compute.
2. En la lista de flotas aprovisionadas, elija la flota que desee eliminar.
3. Elija Eliminar.
4. Introduzca **delete** para confirmar la eliminación.
5. Elija Eliminar.

Asignar una flota aprovisionada o un procesamiento bajo demanda a una acción

De forma predeterminada, las acciones del flujo de trabajo utilizan la flota `Linux.x86-64.Large` bajo demanda con un tipo de cómputo de Amazon EC2. Para usar una flota aprovisionada en su lugar o usar una flota bajo demanda diferente, por ejemplo `Linux.x86-64.2XLarge`, siga las instrucciones siguientes.

Visual

Antes de empezar

- Si desea asignar una flota aprovisionada, primero debe crear la flota aprovisionada. Para obtener más información, consulte [Crear, editar y eliminar una flota aprovisionada](#).

Para asignar una flota aprovisionada o un tipo de flota diferente a una acción

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción a la que desee asignar la flota aprovisionada o el nuevo tipo de flota.
8. Elija la pestaña Configuración.
9. En Compute fleet, haz lo siguiente:

Especifica la máquina o flota que ejecutará tu flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte [Propiedades de la flota aprovisionada](#)

Si `Fleet` se omite, el valor predeterminado es `Linux.x86-64.Large`

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Antes de empezar

- Si desea asignar una flota aprovisionada, primero debe crear la flota aprovisionada. Para obtener más información, consulte [Crear, editar y eliminar una flota aprovisionada](#).

Para asignar una flota aprovisionada o un tipo de flota diferente a una acción

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo. /
5. Elija Editar.
6. Elige YAML.
7. Busca la acción a la que quieres asignar la flota aprovisionada o el nuevo tipo de flota.
8. En la acción, añade una Compute propiedad y configúrala Fleet con el nombre de tu flota o el tipo de flota a pedido. Para obtener más información, consulte la descripción de la Fleet propiedad en la sección [Referencia de acciones de compilación y prueba](#) correspondiente a su acción.
9. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Compartir el cómputo entre las acciones

Temas

- [Ejecutar varias acciones en el cómputo compartido](#)
- [Consideraciones sobre el uso compartido de la computación](#)
- [Activar el uso compartido del cómputo](#)
- [Ejemplos](#)

De forma predeterminada, las acciones de un flujo de trabajo se ejecutan en instancias independientes de una [flota](#). Este comportamiento proporciona a las acciones aislamiento y previsibilidad en cuanto al estado de las entradas. El comportamiento predeterminado requiere una configuración explícita para compartir el contexto, como archivos y variables, entre las acciones.

El uso compartido de recursos informáticos es una capacidad que permite ejecutar todas las acciones de un flujo de trabajo en la misma instancia. El uso del cómputo compartido puede proporcionar tiempos de ejecución del flujo de trabajo más rápidos, ya que se dedica menos tiempo a aprovisionar instancias. También puedes compartir archivos (artefactos) entre acciones sin necesidad de configurar el flujo de trabajo adicional.

Cuando un flujo de trabajo se ejecuta mediante el uso compartido de recursos informáticos, se reserva una instancia de la flota predeterminada o especificada durante todas las acciones de ese flujo de trabajo. Cuando se completa la ejecución del flujo de trabajo, se libera la reserva de instancia.

Ejecutar varias acciones en el cómputo compartido

Puedes usar el `Compute` atributo de la definición YAML en el nivel del flujo de trabajo para especificar las propiedades de las acciones para compartir la flota y el cómputo. También puede configurar las propiedades de procesamiento mediante el editor visual de CodeCatalyst. Para especificar una flota, establece el nombre de una flota existente, establece el tipo de cómputo en EC2 y activa el cómputo compartido.

Note

El uso compartido de recursos solo se admite si el tipo de procesamiento está configurado en EC2 y no es compatible con el sistema operativo Windows Server 2022. Para obtener más información sobre las flotas informáticas, los tipos de cómputo y las propiedades, consulte.

[Trabajar con computación](#)

Note

Si estás en el nivel gratuito y especificas la `Linux.x86-64.2XLarge` flota `Linux.x86-64.XLarge` o la flota manualmente en la definición de flujo de trabajo YAML, la acción seguirá ejecutándose en la flota predeterminada (`Linux.x86-64.Large`). Para

obtener más información sobre la disponibilidad del procesamiento y los precios, consulta la [tabla con las opciones de niveles](#).

Cuando la función de compartir el procesamiento está activada, la carpeta que contiene la fuente del flujo de trabajo se copia automáticamente en todas las acciones. No es necesario configurar los artefactos de salida ni hacer referencia a ellos como artefactos de entrada en una definición de flujo de trabajo (archivo YAML). Como autor de un flujo de trabajo, debe configurar las variables de entorno mediante entradas y salidas, del mismo modo que lo haría sin utilizar el cómputo compartido. Si quieres compartir carpetas entre acciones ajenas a la fuente del flujo de trabajo, considera la posibilidad de almacenar archivos en caché. Para obtener más información, consulte [Trabajar con artefactos](#) y [Trabajar con el almacenamiento en caché de archivos](#).

El repositorio de origen en el que reside el archivo de definición del flujo de trabajo se identifica mediante la etiqueta `WorkflowSource`. Al utilizar el procesamiento compartido, la fuente del flujo de trabajo se descarga en la primera acción que hace referencia a ella y queda automáticamente disponible para que la utilicen las siguientes acciones del flujo de trabajo que se ejecute. Cualquier cambio realizado en la carpeta que contiene la fuente del flujo de trabajo mediante una acción, como añadir, modificar o eliminar archivos, también se verá en las acciones subsiguientes del flujo de trabajo. Puedes hacer referencia a los archivos que se encuentran en la carpeta de origen del flujo de trabajo en cualquiera de tus acciones del flujo de trabajo, del mismo modo que lo harías sin utilizar la compartición de recursos informáticos. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#).

Note

Los flujos de trabajo de cómputo compartido deben especificar una secuencia estricta de acciones, por lo que no se pueden configurar acciones paralelas. Si bien los artefactos de salida se pueden configurar en cualquier acción de la secuencia, no se admiten los artefactos de entrada.

Consideraciones sobre el uso compartido de la computación

Puede ejecutar flujos de trabajo con procesamiento compartido para acelerar las ejecuciones de los flujos de trabajo y compartir el contexto entre las acciones de un flujo de trabajo que utilizan la misma instancia. Ten en cuenta lo siguiente para determinar si el uso del cómputo compartido es adecuado para tu escenario:

	Uso compartido de computación	Sin compartir el cómputo
Tipo de computación	Amazon EC2	Amazon EC2, AWS Lambda
Aprovisionamiento de instancias	Las acciones se ejecutan en la misma instancia	Las acciones se ejecutan en instancias distintas
Sistema operativo	Amazon Linux 2	Amazon Linux 2, Windows Server 2022 (solo acción de compilación)
Archivos de referencia	<code>\$CATALYST_SOURCE_DIRECTORY/WorkflowSource/</code>	<code>\$CATALYST_SOURCE_DIRECTORY/WorkflowSource/</code>
Estructura de flujo de trabajo	Las acciones solo se pueden ejecutar de forma secuencial	Las acciones pueden ejecutarse en paralelo
Acceder a los datos en todas las acciones del flujo	Acceder a la fuente de flujo de trabajo en caché (<code>WorkflowSource</code>)	Acceda a las salidas de artefactos compartidos (requiere una configuración adicional)

Activar el uso compartido del cómputo

Usa las siguientes instrucciones para activar la compartición de cómputo en un flujo de trabajo.

Visual

Para activar el uso compartido del cómputo mediante el editor visual

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo.
5. Elija Editar.

6. Elige Visual.
7. Elija las propiedades del flujo de trabajo.
8. En el menú desplegable Tipo de cómputo, elija EC2.
9. (Opcional) En el menú desplegable Flota de cómputo (opcional), selecciona la flota que quieras usar para ejecutar las acciones del flujo de trabajo. Puedes elegir una flota a pedido o crear y elegir una flota aprovisionada. Para obtener más información, consulte [Crear, editar y eliminar una flota aprovisionada](#) y [Asignar una flota aprovisionada o un procesamiento bajo demanda a una acción](#).
10. Cambia el botón para activar el uso compartido del cómputo y hacer que las acciones del flujo de trabajo se ejecuten en la misma flota.
11. (Opcional) Elige el modo de ejecución del flujo de trabajo. Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#).
12. Seleccione Confirmar, introduzca un mensaje de confirmación y vuelva a seleccionar Confirmar.

YAML

Para activar el uso compartido del cómputo mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Activa el uso compartido del procesamiento configurando el SharedInstance campo en TRUE y Type en EC2. Configure Fleet la flota de procesamiento que desee usar para ejecutar acciones de flujo de trabajo. Puede elegir una flota a pedido o crear y elegir una flota aprovisionada. Para obtener más información, consulte [Crear, editar y eliminar una flota aprovisionada](#) y [Asignar una flota aprovisionada o un procesamiento bajo demanda a una acción](#).

En un flujo de trabajo YAML, agrega un código similar al siguiente:

```
Name: MyWorkflow
SchemaVersion: "1.0"
Compute: # Define compute configuration.
  Type: EC2
  Fleet: MyFleet # Optionally, choose an on-demand or provisioned fleet.
  SharedInstance: true # Turn on compute sharing. Default is False.
Actions:
  BuildFirst:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ...
        ...
```

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Ejemplos

Temas

- [Ejemplo: Amazon S3 Publish](#)

Ejemplo: Amazon S3 Publish

Los siguientes ejemplos de flujo de trabajo muestran cómo realizar la acción de publicación en Amazon S3 de dos maneras: primero mediante artefactos de entrada y, a continuación, mediante el uso compartido de cómputo. Con el cómputo compartido, los artefactos de entrada no son necesarios, ya que se puede acceder a lo almacenado en caché `WorkflowSource`. Además, el artefacto de salida de la acción Construir ya no es necesario. La acción de publicación de S3 está configurada para utilizar la `DependsOn` propiedad explícita para mantener las acciones secuenciales; la acción de creación debe ejecutarse correctamente para que la acción de publicación de S3 se ejecute.

- Sin compartir el cómputo, es necesario usar artefactos de entrada y compartir los resultados con las siguientes acciones:

```
Name: S3PublishUsingInputArtifact
SchemaVersion: "1.0"
Actions:
  Build:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ArtifactToPublish
          Files: [output.zip]
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ./build.sh # Build script that generates output.zip
  PublishToS3:
    Identifier: aws/s3-publish@v1
    Inputs:
      Artifacts:
        - ArtifactToPublish
    Environment:
      Connections:
        - Role: codecatalyst-deployment-role
          Name: dev-deployment-role
      Name: dev-connection
    Configuration:
      SourcePath: output.zip
      DestinationBucketName: dev-bucket
```

- Si utilizas la compartición de recursos con la configuración `SharedInstance` en `TRUE`, puedes ejecutar varias acciones en la misma instancia y compartir artefactos especificando una única fuente de flujo de trabajo. Los artefactos de entrada no son obligatorios y no se pueden especificar:

```
Name: S3PublishUsingComputeSharing
SchemaVersion: "1.0"
Compute:
  Type: EC2
  Fleet: dev-fleet
  SharedInstance: TRUE
```

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ./build.sh # Build script that generates output.zip
  PublishToS3:
    Identifier: aws/s3-publish@v1
    DependsOn:
      - Build
    Environment:
      Connections:
        - Role: codecatalyst-deployment-role
          Name: dev-deployment-role
      Name: dev-connection
    Configuration:
      SourcePath: output.zip
      DestinationBucketName: dev-bucket
```

Trabajar con imágenes de Docker en un entorno de ejecución

Una imagen de entorno de ejecución es un contenedor de Docker en el que CodeCatalyst se ejecutan las acciones del flujo de trabajo. El contenedor Docker se ejecuta sobre la plataforma de procesamiento elegida e incluye un sistema operativo y herramientas adicionales que una acción de flujo de trabajo podría necesitar, como Node.js y tar. AWS CLI

De forma predeterminada, las acciones del flujo de trabajo se ejecutarán en una de las [imágenes activas](#) suministradas y mantenidas por CodeCatalyst. Solo las acciones de creación y prueba admiten imágenes personalizadas. Para obtener más información, consulte [Asignación de una imagen de Docker de un entorno de ejecución personalizado a una acción](#).

Temas

- [Imágenes activas](#)
- [¿Qué pasa si una imagen activa no incluye las herramientas que necesito?](#)
- [Asignación de una imagen de Docker de un entorno de ejecución personalizado a una acción](#)
- [Ejemplos](#)

Imágenes activas

Las imágenes activas son imágenes del entorno de ejecución totalmente compatibles con herramientas preinstaladas CodeCatalyst e incluidas. Actualmente hay dos conjuntos de imágenes activas: uno publicado en marzo de 2024 y otro publicado en noviembre de 2022.

El hecho de que una acción utilice una imagen de marzo de 2024 o de noviembre de 2022 depende de la acción:

- Las acciones de creación y prueba que se agreguen a un flujo de trabajo el 26 de marzo de 2024 o después incluirán una `Container` sección en su definición de YAML que especifique explícitamente una [imagen de marzo de 2024](#). Si lo desea, puede eliminar la `Container` sección para volver a la imagen de [noviembre de 2022](#).
- Las acciones de creación y prueba que se hayan agregado a un flujo de trabajo antes del 26 de marzo de 2024 no incluirán ninguna `Container` sección en su definición de YAML y, por lo tanto, utilizarán una imagen de [noviembre de 2022](#). Puedes conservar la imagen de noviembre de 2022 o puedes actualizarla. Para actualizar la imagen, abra la acción en el editor visual, elija la pestaña Configuración y, a continuación, seleccione la imagen de marzo de 2024 en la lista desplegable de imágenes docker del entorno de ejecución. Esta selección añadirá una `Container` sección a la definición de YAML de la acción que se rellenará con la imagen correspondiente de marzo de 2024.
- Todas las demás acciones utilizarán una [imagen de noviembre de 2022](#), independientemente de cuándo se hayan agregado al flujo de trabajo. Actualmente, no es posible actualizar estas acciones para usar una imagen de marzo de 2024.

Temas

- [Imágenes de marzo de 2024](#)
- [Imágenes de noviembre de 2022](#)

Imágenes de marzo de 2024

Las imágenes de marzo de 2024 son las últimas imágenes proporcionadas por CodeCatalyst. Hay una imagen de marzo de 2024 por combinación de tipo de cómputo y flota.

La siguiente tabla muestra las herramientas instaladas en cada imagen de marzo de 2024.

Herramientas de imagen de marzo de 2024

Herramienta	CodeCatalyst Amazon EC2 para Linux x86_64 - CodeCatalystLinux_x86_64:2024_03	CodeCatalyst Lambda para Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2024_03	CodeCatalyst Amazon EC2 para Linux Arm64 - CodeCatalystLinux_Arm64:2024_03	CodeCatalyst Lambda para Linux Arm64 - CodeCatalystLinuxLambda_Arm64:2024_03
AWS CLI	2.15.17	2.15.17	2.15.17	2.15.17
AWS Copiloto CLI	1.32.1	1,32.1	1,32.1	1,32.1
Docker	240,9	N/A	24.0.9	N/A
Docker Compose	2.23.3	N/A	2.23.3	N/A
Git	2.43,0	2.43,0	2.43,0	2.43,0
Go	1,21,5	1.21.5	1.21.5	1.21.5
Gradle	8.5	8.5	8.5	8.5
Java	Corretto 17	Corretto 17	Corretto 17	Corretto 17
Maven	3.9.6	3.9.6	3.9.6	3.9.6
Node.js	18,19,0	18,19,0	18,19,0	18,19,0
npm	102.3	10.2.3	10.2.3	10.2.3
Python	3,9,18	3.9,18	3.9,18	3.9,18
Python3	3.11.6	3.11.6	3.11.6	3.11.6
pip	22.3.1	22.3.1	22.3.1	22.3.1
.NET	8.0.100	8.0.100	8.0.100	8.0.100

Imágenes de noviembre de 2022

Hay una imagen de noviembre de 2022 por combinación de tipo de cómputo y flota. También hay una imagen de Windows de noviembre de 2022 disponible con la acción de compilación si ha configurado una flota [aprovisionada](#).

En la siguiente tabla se muestran las herramientas instaladas en cada imagen de noviembre de 2022.

Herramientas de imagen de noviembre de 2022

Herramienta	CodeCatalyst Amazon EC2 para Linux x86_64 - CodeCatalystLinux_x86_64:2022_11	CodeCatalyst Lambda para Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2022_11	CodeCatalyst Amazon EC2 para Linux Arm64 - CodeCatalystLinux_Arm64:2022_11	CodeCatalyst Lambda para Linux Arm64 - CodeCatalystLinuxLambda_Arm64:2022_11
AWS CLI	2.15.17	2.15.17	2.15.17	2.15.17
AWS Copiloto CLI	0.6.0	0.6.0	N/A	N/A
Docker	23,01	N/A	23.0.1	N/A
Docker Compose	2.16.0	N/A	2.16.0	N/A
Git	2.40.0	2.40.0	2.39,2	2.39,2
Go	1.20.2	1.20.2	1.20.1	1,20.1
Gradle	8.0.2	8.0.2	8.0.1	8.0.1
Java	Corretto 17	Corretto 17	Corretto 17	Corretto 17
Maven	3.9.4	3.9.4	3.9.0	3.9.0
Node.js	16,20,2	16,20,2	16,19,1	16,14.2
npm	8,19,4	8.19,4	8.19,3	8.5.0
Python	3,9,15	2.7.18	3.11.2	2.7.18

Herramienta	CodeCatalyst Amazon EC2 para Linux x86_64 - CodeCatalystLinux_ x86_64:2022_11	CodeCatalyst Lambda para Linux x86_64 - CodeCatalystLinuxL ambda_x86 _64:2022_11	CodeCatalyst Amazon EC2 para Linux Arm64 - CodeCatalystLinux_ Arm64:2022_11	CodeCatalyst Lambda para Linux Arm64 - CodeCatalystLinux_ ambda_Ar 64:2022_11
Python3	N/A	3,9,15	N/A	3.11.2
pip	22.2.2	22.2.2	23.0.1	23.0.1
.NET	6.0.407	6,0407	6,0406	6.0.406

¿Qué pasa si una imagen activa no incluye las herramientas que necesito?

Si ninguna de las [imágenes activas](#) suministradas por CodeCatalyst incluye las herramientas que necesita, tiene un par de opciones:

- Puede proporcionar una imagen de Docker de un entorno de ejecución personalizado que incluya las herramientas necesarias. Para obtener más información, consulte [Asignación de una imagen de Docker de un entorno de ejecución personalizado a una acción](#).

Note

Si deseas proporcionar una imagen de Docker de un entorno de ejecución personalizado, asegúrate de que la imagen personalizada tenga Git instalado.

- Puedes hacer que la acción de creación o prueba de tu flujo de trabajo instale las herramientas que necesites.

Por ejemplo, puedes incluir las siguientes instrucciones en la Steps sección del código YAML de la acción de compilación o prueba:

```
Configuration:
Steps:
- Run: ./setup-script
```

A continuación, la instrucción *setup-script* ejecutaría el siguiente script para instalar el administrador de paquetes Node (npm):

```
#!/usr/bin/env bash
echo "Setting up environment"

touch ~/.bashrc
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install v16.1.0
source ~/.bashrc
```

Para obtener más información sobre la acción de compilación YAML, consulte [Referencia de acciones de compilación y prueba](#)

Asignación de una imagen de Docker de un entorno de ejecución personalizado a una acción

Si no quieres usar una [imagen activa proporcionada por CodeCatalyst, puedes proporcionar una imagen](#) de Docker de un entorno de ejecución personalizado. Si quieres proporcionar una imagen personalizada, asegúrate de que tenga Git instalado. La imagen puede residir en Docker Hub, Amazon Elastic Container Registry o cualquier repositorio público.

Para obtener información sobre cómo crear una imagen de Docker personalizada, consulte [Containerizar una aplicación en contenedores](#) en la documentación de Docker.

Usa las siguientes instrucciones para asignar una imagen de Docker de tu entorno de ejecución personalizado a una acción. Tras especificar una imagen, la CodeCatalyst despliega en la plataforma de procesamiento cuando se inicia la acción.

Note

Las siguientes acciones no son compatibles con las imágenes de Docker del entorno de ejecución personalizado: Deploy AWS CloudFormation stack, Deploy to ECS y GitHub Actions. Las imágenes de Docker del entorno de ejecución personalizado tampoco admiten el tipo de cálculo Lambda.

Visual

Para asignar un entorno de ejecución personalizado, una imagen de Docker mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
4. Elija Editar.
5. Elija Visual.
6. En el diagrama de flujo de trabajo, elige la acción que utilizará la imagen de Docker de tu entorno de ejecución personalizado.
7. Elija la pestaña Configuración.
8. En la parte inferior, rellena los siguientes campos.

Imagen de Docker del entorno de ejecución: opcional

Especifique el registro en el que se almacena la imagen. Los valores válidos son:

- CODECATALYST(editor YAML)

La imagen se guarda en el CodeCatalyst registro.

- Docker Hub (editor visual) o DockerHub (editor YAML)

La imagen se almacena en el registro de imágenes de Docker Hub.

- Otro registro (editor visual) o Other (editor YAML)

La imagen se guarda en un registro de imágenes personalizado. Se puede utilizar cualquier registro disponible públicamente.

- Amazon Elastic Container Registry (editor visual) o ECR (editor YAML)

La imagen se almacena en un repositorio de imágenes de Amazon Elastic Container Registry. Para usar una imagen en un repositorio de Amazon ECR, esta acción necesita acceso a Amazon ECR. Para habilitar este acceso, debe crear un [rol de IAM](#) que incluya los siguientes permisos y una política de confianza personalizada. (Si lo desea, puede [modificar un rol existente para incluir los permisos y la política](#)).

El rol de IAM debe incluir los siguientes permisos en su política de roles:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

El rol de IAM debe incluir la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obtener más información sobre la creación de funciones de IAM, consulte [Creación de una función mediante políticas de confianza personalizadas \(consola\)](#) en la Guía del usuario de IAM.

Una vez que haya creado el rol, debe asignarlo a la acción a través de un entorno. Para obtener más información, consulte [Asociar un entorno, una conexión de cuenta y un rol de IAM a una acción de flujo de trabajo](#).

URL de imagen ECR, imagen de Docker Hub o URL de imagen

Especifique uno de los siguientes valores:

- Si utiliza un CODECATALYST registro, defina la imagen en una de las siguientes imágenes [activas](#):
 - CodeCatalystLinux_x86_64:2024_03
 - CodeCatalystLinux_x86_64:2022_11
 - CodeCatalystLinux_Arm64:2024_03
 - CodeCatalystLinux_Arm64:2022_11
 - CodeCatalystLinuxLambda_x86_64:2024_03
 - CodeCatalystLinuxLambda_x86_64:2022_11
 - CodeCatalystLinuxLambda_Arm64:2024_03
 - CodeCatalystLinuxLambda_Arm64:2022_11
 - CodeCatalystWindows_x86_64:2022_11
- Si utiliza un registro de Docker Hub, defina la imagen con el nombre de la imagen de Docker Hub y la etiqueta opcional.

Ejemplo: `postgres:latest`

- Si utiliza un registro de Amazon ECR, defina la imagen en el URI del registro de Amazon ECR.

Ejemplo: `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Si utiliza un registro personalizado, defina la imagen en el valor esperado por el registro personalizado.

9. (Opcional) Elija Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para asignar una imagen de Docker de un entorno de ejecución personalizado mediante el editor YAML

1. En el panel de navegación, selecciona CI/CD y, a continuación, selecciona Flujos de trabajo.
2. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

3. Elija Editar.
4. Elige YAML.
5. Busca la acción a la que quieres asignar una imagen de Docker de un entorno de ejecución.
6. En la acción, añade una Container sección Registry y Image sus propiedades subyacentes. Para obtener más información, consulte la Container descripción Registry y Image las propiedades [Acciones](#) de la acción.
7. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
8. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Ejemplos

Los siguientes ejemplos muestran cómo asignar una imagen de Docker de un entorno de ejecución personalizado a una acción del archivo de definición del flujo de trabajo.

Temas

- [Ejemplo: uso de una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con Amazon ECR](#)
- [Ejemplo: usar una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con Docker Hub](#)

Ejemplo: uso de una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con Amazon ECR

El siguiente ejemplo muestra cómo utilizar una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con [Amazon ECR](#).

```
Configuration:
  Container:
    Registry: ECR
    Image: public.ecr.aws/amazonlinux/amazonlinux:2023
```

Ejemplo: usar una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con Docker Hub

[En el siguiente ejemplo, se muestra cómo utilizar una imagen de Docker de un entorno de ejecución personalizado para añadir compatibilidad con Node.js 18 con Docker Hub.](#)

```
Configuration:
  Container:
    Registry: DockerHub
    Image: node:18.18.2
```

Trabajo con entornos

Un entorno, que no debe confundirse con un [entorno de desarrollo](#), es donde se despliega el código. Por lo general, contiene una instancia de una aplicación en ejecución junto con su infraestructura asociada. Puede asignar un nombre a su entorno, como desarrollo, prueba, puesta en escena o producción. Todas las implementaciones generadas por CodeCatalyst un entorno aparecerán en la página Entornos. Para configurar un entorno, debe asignarle un nombre, por ejemplo `my-production-environment`, y luego asociarlo al suyoCuenta de AWS.

Además de mostrar la información de implementación, un entorno también sirve como mecanismo mediante el cual se asignan funciones de AWS IAM a [las acciones](#) del flujo de trabajo.

¿Pueden existir varios entornos dentro de un único flujo de trabajo?

Sí. Si un flujo de trabajo incluye varias acciones, a cada una de ellas se le puede asignar un entorno. Por ejemplo, puede tener un flujo de trabajo que incluya dos acciones de despliegue, en las que a una se le asigne un `my-staging-environment` entorno y a otra se le asigne un `my-production-environment` entorno.

¿Qué acciones son compatibles con los entornos?

Las siguientes acciones permiten que su información de despliegue se muestre en la página de entornos:

- Pila AWS CloudFormation de implementación: para obtener más información, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#)
- Implementación en Amazon ECS: para obtener más información, consulte [Añadir la acción «Implementar en Amazon ECS»](#)

- Implementación en un clúster de Kubernetes: para obtener más información, consulte [Añadir la acción «Implementar en un clúster de Kubernetes»](#)
- AWS CDK Implementar: para obtener más información, consulte [Añadir la acción «AWS CDK desplegar»](#)

Note

Si quieres permitir que una acción acceda a tu AWS cuenta y realice operaciones en ella, tendrás que asociarla a un entorno. Muchas acciones apoyan a una asociación medioambiental, incluidas, entre otras, las acciones enumeradas anteriormente. Puede saber qué acciones son compatibles con una asociación de entornos porque incluirán una lista desplegable de entornos en la pestaña Configuración del [editor visual](#).

Regiones admitidas

La página de entornos puede mostrar los recursos de cualquier AWS región.

¿Los entornos son obligatorios?

Un entorno es obligatorio si la acción de flujo de trabajo a la que está asignado despliega recursos en la AWS nube o se comunica con AWS los servicios por otros motivos (como la supervisión y la generación de informes).


Creación de un entorno

Siga las instrucciones siguientes para crear un entorno vacío que pueda asociar posteriormente a una acción de flujo de trabajo.

Antes de empezar

Necesita lo siguiente:

- Un CodeCatalyst espacio. Para obtener más información, consulte [Configuración CodeCatalyst](#).
- Un CodeCatalyst proyecto. Para obtener más información, consulte [Crear un proyecto con un plano](#).
- Una conexión de AWS cuenta que incluye las funciones de IAM a las que necesitará acceder AWS tu acción de flujo de trabajo. Puede utilizar como máximo una conexión de cuenta por entorno. Para obtener más información, consulte [Cuentas de AWS Administrar un espacio](#).

 Note

Puede crear un entorno sin una conexión de cuenta; sin embargo, tendrá que volver y añadir la conexión más adelante.

Para crear una de entorno

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
4. En Nombre del entorno, introduzca un nombre, como **Production** o **Staging**.
5. En Tipo de entorno, seleccione una de las siguientes opciones:
 - Fuera de producción: un entorno en el que puede probar su aplicación para asegurarse de que funciona según lo previsto antes de pasarla a producción.
 - Producción: un entorno «en vivo» que está disponible públicamente y que aloja la aplicación finalizada.

Si eliges Producción, aparecerá un distintivo de Producción en la interfaz de usuario junto a cualquier acción a la que esté asociado el entorno. El distintivo le ayuda a ver rápidamente qué acciones se están implementando en la producción. Aparte de la apariencia de la insignia, no hay diferencias entre los entornos de producción y los que no lo son.

6. (Opcional) En la conexión de VPC, elija una conexión de VPC que desee asociar a este entorno. Para obtener más información sobre la creación de esta conexión de VPC, consulte [Gestión de Amazon Virtual Private Clouds](#) en la Guía del CodeCatalyst administrador.
7. (Opcional) En Descripción, introduzca una descripción como **Production environment for the hello-world app**.
8. En Conexión: opcional, elija la conexión de AWS cuenta que desee asociar a este entorno. Asegúrese de que la conexión de la cuenta incluya las funciones de IAM que desee asociar al entorno. Para obtener más información sobre la creación de esta conexión, consulte [Cuentas de AWS Administrar un espacio](#).
9. Elija Crear entorno. CodeCatalyst crea un entorno vacío.

Siguientes pasos

- Ahora que ha creado un entorno, está listo para asociarlo a una acción de flujo de trabajo. Para obtener más información, consulte [Asociar un entorno, una conexión de cuenta y un rol de IAM a una acción de flujo de trabajo](#).

Asociar un entorno, una conexión de cuenta y un rol de IAM a una acción de flujo de trabajo

Al asociar un entorno, una conexión de cuentas y un rol de IAM a una [acción de flujo de trabajo compatible](#), el rol de IAM pasa a estar disponible para que lo utilice la acción. Además de acceder a un rol de IAM, es posible que la información de implementación de la acción también se importe a la página de entornos. Para obtener más información, consulte [¿Qué acciones son compatibles con los entornos?](#).

Siga las instrucciones siguientes para asociar un entorno, una conexión de cuenta y un rol de IAM a una acción.

Paso 1: asocie el entorno, la conexión de la cuenta y el rol a una acción del flujo de trabajo

Utilice el siguiente procedimiento para asociar un entorno, una conexión de cuenta y un rol a una acción de flujo de trabajo.

Visual

Para asociar un entorno, una cuenta, una conexión y un rol a una acción de flujo de trabajo mediante el editor visual

- Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
- Elija el proyecto.
- En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
- Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
- Elija Editar.
- Elija Visual.
- En el diagrama de flujo de trabajo, elija una acción que sea compatible con los entornos. Para obtener más información, consulte [¿Qué acciones son compatibles con los entornos?](#).

8. Seleccione la pestaña Configuración y especifique la información en los campos, de la siguiente manera.

Entorno

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Conexión a la cuenta o Conexión: opcional (la que esté disponible)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo `Environment`.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Rol

Especifique el nombre de la función de IAM que utiliza esta acción para acceder y operar en AWS servicios como Amazon S3 y Amazon ECR. Asegúrese de añadir esta función a la conexión de su cuenta. Para añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Es posible que pueda especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí, siempre que tenga permisos suficientes. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol para su cuenta y su espacio](#). Tenga en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Te recomendamos que solo utilices este rol en los tutoriales y en situaciones en las que la seguridad sea un problema menor.

Si no ve el rol en la lista, es porque no lo ha asociado a la conexión de la cuenta. Para obtener más información, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

9. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
10. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para asociar un entorno, una cuenta, una conexión y un rol a una acción de flujo de trabajo mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la acción de flujo de trabajo que desee asociar a un entorno, añada un código similar al siguiente:

```
action-name
Environment:
  Name: environment-name
Connections:
  - Name: account-connection-name
    Role: iam-role-name
```

Para obtener más información, consulte la [Referencia de definición de flujo de trabajo](#) sección para su acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.

9. Seleccione Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Paso 2: poblar el entorno

Tras asociar un entorno, una conexión de cuenta y un rol a la acción del flujo de trabajo, puede rellenar la página de entornos con información sobre el despliegue. Siga las instrucciones siguientes para rellenar la página de entornos.

Note

La página Entornos solo es compatible con un subconjunto de acciones del flujo de trabajo. Para obtener más información, consulte [¿Qué acciones son compatibles con los entornos?](#).

Para poblar el entorno

1. Si la ejecución de un flujo de trabajo no se inició automáticamente al realizar los cambios [Paso 1: asocie el entorno, la conexión de la cuenta y el rol a una acción del flujo de trabajo](#), inicie una ejecución manualmente de la siguiente manera:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija el nombre del flujo de trabajo en el que desee iniciar una ejecución. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 - c. Elija Ejecutar.

La ejecución del flujo de trabajo inicia una nueva implementación, lo que hace que se agregue CodeCatalyst la información sobre los recursos de la aplicación en Entornos.

2. Compruebe que los recursos de la aplicación aparecen en su entorno:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
 - b. Elija su entorno (por ejemplo, `Production`).
 - c. Seleccione la pestaña Actividad de despliegue y compruebe que el despliegue aparezca con el estado EXITOSO. Esto indica que la ejecución de un flujo de trabajo ha desplegado correctamente los recursos de la aplicación.

- d. Seleccione la pestaña Objetivos de despliegue y compruebe que aparecen los recursos de la aplicación.

Gestión de un entorno

Utilice las siguientes instrucciones para administrar un entorno asociándolo a una conexión de VPC o a una Cuenta de AWS

Asociación de una conexión de VPC a un entorno

Cuando se configura una acción con un entorno que tiene una conexión de VPC, la acción se ejecutará conectada a la VPC, respetando las reglas de red y los recursos de acceso especificados por la VPC asociada. Uno o más entornos pueden utilizar la misma conexión de VPC.

Siga las instrucciones siguientes para asociar una conexión de VPC a un entorno.

Para asociar una conexión de VPC a un entorno

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
4. Elija su entorno (por ejemplo, `Production`).
5. Elija la pestaña de propiedades del entorno.
6. Elija Administrar conexión de VPC, elija la conexión de VPC que desee y elija Confirmar. Esto asocia la conexión de VPC seleccionada a este entorno.

Para obtener más información, consulte [Gestión de Amazon Virtual Private Clouds](#) en la Guía CodeCatalyst del administrador.

Asociación de Cuenta de AWS un entorno

Utilice las siguientes instrucciones para Cuenta de AWS asociar un a un entorno.

Para asociar un Cuenta de AWS a un entorno

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.

3. En el panel de navegación, elija CI/CD y, a continuación, elija Entornos.
4. Elija su entorno (por ejemplo, `Production`).
5. Elija la pestaña de propiedades del entorno.
6. Elija Asociar Cuenta de AWS, elija lo que desee Cuenta de AWS y elija Asociar. Esto asociará su selección Cuenta de AWS con este entorno.

Para obtener más información, consulte [Cuentas de AWS Administrar un espacio](#).

Trabajar con el almacenamiento en caché de archivos

Cuando el almacenamiento en caché de archivos está activado, las acciones de compilación y prueba guardan los archivos del disco en una memoria caché y los restauran desde esa memoria caché en las siguientes ejecuciones del flujo de trabajo. El almacenamiento en caché reduce la latencia provocada por la creación o descarga de dependencias que no han cambiado entre ejecuciones. CodeCatalyst también es compatible con las cachés alternativas, que se pueden usar para restaurar cachés parciales que contienen algunas de las dependencias necesarias. Esto ayuda a reducir el impacto en la latencia de una pérdida de memoria caché.

Note

El almacenamiento en caché de archivos solo está disponible con las acciones de CodeCatalyst [compilación](#) y [prueba](#) de Amazon, y solo cuando están configuradas para usar el tipo de [procesamiento](#) EC2.

Temas

- [Acerca del almacenamiento en caché de archivos](#)
- [Crear una memoria caché](#)
- [Restricciones](#)

Acerca del almacenamiento en caché de archivos

El almacenamiento en caché de archivos permite organizar los datos en varias cachés, a las que se hace referencia en la propiedad. `FileCaching` Cada caché guarda un directorio especificado por una ruta determinada. El directorio especificado se restaurará en futuras ejecuciones del flujo de

trabajo. A continuación, se muestra un ejemplo de fragmento de código YAML para almacenar en caché con varias cachés denominadas y. cacheKey1 cacheKey2

```
Actions:
  BuildMyNpmApp:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: npm install
        - Run: npm run test
    Caching:
      FileCaching:
        cacheKey1:
          Path: file1.txt
          RestoreKeys:
            - restoreKey1
        cacheKey2:
          Path: /root/repository
          RestoreKeys:
            - restoreKey2
            - restoreKey3
```

Note

CodeCatalyst utiliza el almacenamiento en caché multicapa, que consta de una caché local y una caché remota. Cuando las flotas aprovisionadas o las máquinas bajo demanda detectan una falta de caché en una caché local, las dependencias se restaurarán desde una caché remota. Como resultado, es posible que algunas acciones ejecutadas experimenten latencia debido a la descarga de una caché remota.

CodeCatalyst aplica restricciones de acceso a la caché para garantizar que una acción de un flujo de trabajo no pueda modificar las cachés de un flujo de trabajo diferente. Esto protege cada flujo de trabajo de otros que podrían enviar datos incorrectos y afectar a las compilaciones o despliegues. Las restricciones se aplican con alcances de caché que aíslan las cachés para cada flujo de trabajo y emparejamiento de sucursales. Por ejemplo, `workflow-A` una sucursal `feature-A` tiene una caché de archivos diferente a la de una sucursal hermana. `workflow-A feature-B`

Los errores de caché se producen cuando un flujo de trabajo busca una caché de archivos específica y no puede encontrarla. Esto puede ocurrir por varios motivos, como cuando se crea una nueva rama o cuando se hace referencia a una nueva caché y aún no se ha creado. También puede ocurrir cuando una caché caduca, lo que de forma predeterminada ocurre 14 días después de su último uso. Para mitigar las pérdidas de memoria caché y aumentar la tasa de visitas a la memoria caché, CodeCatalyst es compatible con las cachés alternativas. Las cachés de reserva son cachés alternativas y ofrecen la oportunidad de restaurar las cachés parciales, que pueden ser una versión anterior de una caché. Para restaurar una memoria caché, primero se busca una coincidencia con el nombre de la propiedad y, si no se encuentra, se evalúa. `FileCaching RestoreKeys` Si hay una falta de caché tanto para el nombre de la propiedad como para `todosRestoreKeys`, el flujo de trabajo seguirá ejecutándose, ya que el almacenamiento en caché es lo mejor y no está garantizado.

Crear una memoria caché

Puede utilizar las siguientes instrucciones para añadir una memoria caché a su flujo de trabajo.

Visual

Para añadir una caché mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elige la acción a la que quieres añadir la caché.
8. Elija Configuration (Configuración).
9. En Almacenamiento de archivos en caché (opcional), selecciona Añadir caché e introduce la información en los campos de la siguiente manera:

Clave

Especifique el nombre de la propiedad de caché principal. Los nombres de las propiedades de la caché deben ser únicos en el flujo de trabajo. Cada acción puede tener hasta cinco entradas `FileCaching`.

Ruta

Especifique la ruta asociada a la memoria caché.

Restaurar claves: opcional

Especifique la clave de restauración para utilizarla como alternativa cuando no se encuentre la propiedad principal de la caché. Los nombres de las claves de restauración deben ser exclusivos de su flujo de trabajo. Cada caché puede contener hasta cinco entradas `RestoreKeys`.

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y, a continuación, vuelve a seleccionar Confirmar.

YAML

Para añadir una caché mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una acción de flujo de trabajo, añada un código similar al siguiente:

```
action-name:
  Configuration:
    Steps: ...
  Caching:
    FileCaching:
      key-name:
        Path: file-path
        # # Specify any additional fallback caches
        # RestoreKeys:
```

```
# - restore-key
```

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Restricciones

Las siguientes son las restricciones para el nombre de la propiedad `yRestoreKeys`:

- Los nombres deben ser únicos dentro de un flujo de trabajo.
- Los nombres se limitan a caracteres alfanuméricos (A-Z, a-z, 0-9), guiones (-) y guiones bajos (_).
- Los nombres pueden tener hasta 180 caracteres.
- Cada acción puede tener hasta cinco cachés. `FileCaching`
- Cada caché puede tener hasta cinco entradas. `RestoreKeys`

Las siguientes son las restricciones de las rutas:

- No se permiten los asteriscos (*).
- Las rutas pueden tener un máximo de 255 caracteres.

Trabajar con paquetes

Un paquete es un paquete que incluye el software y los metadatos necesarios para instalar el software y resolver cualquier dependencia. CodeCatalyst admite el formato de paquete npm.

Un paquete se compone de:

- Un nombre (por ejemplo, `webpack` es el nombre de un paquete npm popular)
- Un espacio de [nombres](#) opcional (por ejemplo, en) `@types @types/node`
- Un conjunto de [versiones](#) (por ejemplo, `1.0.0, 1.0.1`) `1.0.2`
- Metadatos a nivel de paquete (por ejemplo, etiquetas npm dist)

En CodeCatalyst, puede publicar paquetes y consumir paquetes de los repositorios de CodeCatalyst paquetes de sus flujos de trabajo. Puedes configurar una acción de compilación o prueba con un

repositorio de CodeCatalyst paquetes para configurar automáticamente el cliente npm de una acción para enviar y extraer paquetes del repositorio especificado.

Para obtener más información sobre los paquetes, consulte [Paquetes en CodeCatalyst](#).

Note

Actualmente, las acciones de compilación y prueba son compatibles con los repositorios de CodeCatalyst paquetes.

Temas

- [Uso de repositorios de CodeCatalyst paquetes en los flujos de trabajo](#)
- [Ejemplos del uso de paquetes en flujos de trabajo](#)

Uso de repositorios de CodeCatalyst paquetes en los flujos de trabajo

En CodeCatalyst, puedes añadir un repositorio de CodeCatalyst paquetes a las acciones de compilación y prueba de tu flujo de trabajo. El repositorio de paquetes debe estar configurado con un formato de paquete, como npm. También puede optar por incluir una secuencia de ámbitos para el repositorio de paquetes seleccionado.

Utilice las siguientes instrucciones para especificar una configuración de paquete para utilizarla con una acción de flujo de trabajo.

Visual

Para especificar la configuración de paquete que utilizará una acción (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.

7. En el diagrama de flujo de trabajo, elija la acción con la que quiere configurar un repositorio de paquetes.
8. Elija Paquetes.
9. En el menú desplegable Añadir configuración, elige la configuración de paquete que quieres usar con las acciones de tu flujo de trabajo.
10. Selecciona Añadir repositorio de paquetes.
11. En el menú desplegable del repositorio de paquetes, especifique el nombre del repositorio de CodeCatalyst paquetes que desea que utilice la acción.

Para obtener más información sobre los repositorios de paquetes, consulte [Repositorios de paquetes](#)

12. (Opcional) En Ámbitos: opcional, especifique la secuencia de ámbitos que desee definir en el registro de paquetes.

Al definir los ámbitos, el repositorio de paquetes especificado se configura como el registro de todos los ámbitos de la lista. Si se solicita un paquete con ese alcance a través del cliente npm, utilizará ese repositorio en lugar del predeterminado. El nombre de cada ámbito debe tener el prefijo «@».

Si Scopes se omite, el repositorio de paquetes especificado se configura como registro predeterminado para todos los paquetes utilizados por la acción.

Para obtener más información sobre los ámbitos, consulte Paquetes [Espacios de nombres de paquetes](#) con [ámbito](#).

13. Elija Add (Agregar).
14. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
15. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para especificar la configuración del paquete que utilizará una acción (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.

3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una acción, añade un código similar al siguiente:

```
action-name:
  Configuration:
    Packages:
      NpmConfiguration:
        PackageRegistries:
          - PackagesRepository: package-repository
        Scopes:
          - "@scope"
```

Para obtener más información, consulte la descripción de la Packages propiedad en la que [Referencia de acciones de compilación y prueba](#) desea realizar la acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Ejemplos del uso de paquetes en flujos de trabajo

Los siguientes ejemplos muestran cómo hacer referencia a los paquetes en el archivo de definición del flujo de trabajo.

Temas

- [Ejemplo: definir paquetes con NpmConfiguration](#)
- [Ejemplo: anular el registro predeterminado](#)
- [Ejemplo: anular los ámbitos del registro de paquetes](#)

Ejemplo: definir paquetes con **NpmConfiguration**

El siguiente ejemplo muestra cómo definir un paquete `NpmConfiguration` en el archivo de definición de flujo de trabajo.

```
Actions:
  Build:
    Identifier: aws/build-beta@v1
    Configuration:
      Packages:
        NpmConfiguration:
          PackageRegistries:
            - PackagesRepository: main-repo
            - PackagesRepository: scoped-repo
          Scopes:
            - "@scope1"
```

En este ejemplo, se configura el cliente `npm` de la siguiente manera:

```
default: main-repo
@scope1: scoped-repo
```

En este ejemplo, hay dos repositorios definidos. El registro predeterminado se establece tal `main-repo` como se definió sin un ámbito. `@scope1`El alcance está configurado `PackageRegistries` `scoped-repo`.

Ejemplo: anular el registro predeterminado

En el siguiente ejemplo, se muestra cómo anular el registro predeterminado.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-repo-1
    - PackagesRepository: my-repo-2
    - PackagesRepository: my-repo-3
```

En este ejemplo, se configura el cliente `npm` de la siguiente manera:

```
default: my-repo-3
```


Si especifica varios repositorios predeterminados, el último repositorio tendrá prioridad. En este ejemplo, el último repositorio de la lista es `my-repo-3`, lo que significa que `npm` se conectará a `my-repo-3`. Esto anula los repositorios `my-repo-1` y `my-repo-2`.

Ejemplo: anular los ámbitos del registro de paquetes

El siguiente ejemplo muestra cómo anular un ámbito en el registro de paquetes.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-default-repo
    - PackagesRepository: my-repo-1
  Scopes:
    - "@scope1"
    - "@scope2"
  - PackagesRepository: my-repo-2
  Scopes:
    - "@scope2"
```

En este ejemplo, se configura el cliente `npm` de la siguiente manera:

```
default: my-default-repo
@scope1: my-repo-1
@scope2: my-repo-2
```

Si incluye ámbitos principales, el último repositorio tendrá prioridad. En este ejemplo, la última vez que `@scope2` se configuró el ámbito es para `PackageRegistries my-repo-2`. Esto anula el ámbito para `my-repo-1` el que `@scope2` se configuró.

Trabajar con corridas

Una ejecución es una iteración única de un flujo de trabajo. Durante una ejecución, CodeCatalyst realiza las acciones definidas en el archivo de configuración del flujo de trabajo y genera los registros, artefactos y variables asociados.

Temas

- [Iniciar una ejecución de flujo de trabajo](#)
- [Detener una ejecución de flujo de trabajo](#)
- [Ver el estado y los detalles de la ejecución del flujo de trabajo](#)
- [Configuración de ejecuciones en cola, reemplazadas y paralelas](#)

Iniciar una ejecución de flujo de trabajo

Utilice el siguiente procedimiento para iniciar la ejecución manual de un flujo de trabajo.

Note

También puede iniciar la ejecución automática de un flujo de trabajo [configurando un activador](#).

Para iniciar un flujo de trabajo, ejecútelo manualmente

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre del flujo de trabajo que desee ejecutar. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Run (Ejecutar).

Detener una ejecución de flujo de trabajo

Utilice el siguiente procedimiento para detener la ejecución de un flujo de trabajo en curso. Es posible que desee detener una ejecución si se inició por accidente.

Al detener una ejecución de un flujo de trabajo, CodeCatalyst espera a que se completen las acciones en curso antes de marcar la ejecución como detenida en la CodeCatalyst consola. Las acciones que no hayan podido iniciarse no se iniciarán y se marcarán como abandonadas.

Note

Si una ejecución está en cola (es decir, no tiene ninguna acción en curso), la ejecución se detiene inmediatamente.

Para detener la ejecución de un flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. En Flujos de trabajo, elija Ejecuciones y elija la ejecución en curso de la lista.
5. Elija Detener.

Ver el estado y los detalles de la ejecución del flujo de trabajo

Puede ver el estado y los detalles de una sola ejecución del flujo de trabajo o de varias ejecuciones al mismo tiempo.

Note

También puede ver el estado del flujo de trabajo, que es diferente del estado de ejecución del flujo de trabajo. Para obtener más información, consulte [Visualización del estado del flujo de trabajo](#).

Temas

- [Estados de ejecución del flujo de trabajo](#)
- [Ver el estado y los detalles de una sola ejecución](#)
- [Ver el estado y los detalles de todas las ejecuciones del proyecto](#)
- [Ver el estado y los detalles de todas las ejecuciones de un flujo de trabajo específico](#)
- [Visualización de las ejecuciones de un flujo de trabajo en el diagrama de flujo de trabajo](#)

Estados de ejecución del flujo de trabajo

La ejecución de un flujo de trabajo puede tener uno de los siguientes estados:

- **Correcto:** la ejecución del flujo de trabajo se procesó correctamente.
- **Error:** se produjo un error en la ejecución de una o más acciones del flujo de trabajo.
- **En curso:** la ejecución del flujo de trabajo se está procesando actualmente.
- **Detenido:** una persona detuvo la ejecución del flujo de trabajo mientras estaba en curso.
- **Deteniéndose:** la ejecución del flujo de trabajo está detenida en este momento.

- **Cancelada:** la ejecución del flujo de trabajo se canceló CodeCatalyst porque el flujo de trabajo asociado se eliminó o actualizó mientras la ejecución estaba en curso.
- **Sustituido:** solo se produce si ha configurado el modo de [ejecución sustituido](#). La ejecución del flujo de trabajo se canceló CodeCatalyst porque una ejecución de flujo de trabajo posterior la reemplazó.

Ver el estado y los detalles de una sola ejecución

Es posible que desee ver el estado y los detalles de una sola ejecución de flujo de trabajo para comprobar si se ha ejecutado correctamente, a qué hora se ha completado o quién o qué la ha iniciado.

Para ver el estado y los detalles de una sola ejecución

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. En el nombre del flujo de trabajo, selecciona Ejecuciones.
6. En Historial de ejecuciones, en la columna ID de ejecución, selecciona una ejecución. Por ejemplo, Run-95a4d .
7. Bajo el nombre de la carrera, realiza una de las siguientes acciones:
 - Visualice un diagrama de flujo de trabajo que muestre las acciones de la ejecución del flujo de trabajo y su estado (consulte [Estados de ejecución del flujo de trabajo](#)). Esta vista también muestra el repositorio y la rama de origen utilizados durante la ejecución.

En el diagrama de flujo de trabajo, elija una acción para ver detalles como los registros, los informes y los resultados generados por la acción durante la ejecución. La información que se muestra depende del tipo de acción que se seleccione. Para obtener más información sobre cómo ver los registros de creación o despliegue, consulte [Visualización de los resultados de una acción de construcción](#) o [Visualización de los registros de despliegue](#).

- **YAML** para ver el archivo de definición del flujo de trabajo que se utilizó para la ejecución.
- **Artefactos** para ver los artefactos producidos por la ejecución del flujo de trabajo. Para obtener más información acerca de los artefactos, consulte [Trabajar con artefactos](#).

- Informes para ver los informes de las pruebas y otros tipos de informes producidos por la ejecución del flujo de trabajo. Para obtener más información acerca de los informes, consulte [Tipos de informes de prueba](#).
- Variables para ver las variables de salida generadas por la ejecución del flujo de trabajo. Para obtener más información sobre las variables, consulte [Trabajar con variables](#).

 Note

Si se ha eliminado el flujo de trabajo principal de la ejecución, aparece un mensaje que indica este hecho en la parte superior de la página de detalles de la ejecución.

Ver el estado y los detalles de todas las ejecuciones del proyecto

Es posible que desee ver el estado y los detalles de todas las ejecuciones del flujo de trabajo del proyecto, comprender la cantidad de actividad del flujo de trabajo que se está produciendo en el proyecto y obtener información sobre el estado general de los flujos de trabajo.

Para ver el estado y los detalles de todas las ejecuciones del proyecto


1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. En Flujos de trabajo, elija Ejecuciones.

Se muestran todas las ejecuciones de todos los flujos de trabajo, en todas las ramas y en todos los repositorios del proyecto.

La página incluye las siguientes columnas:

- ID de ejecución: el identificador único de la ejecución. Elija el enlace del ID de ejecución para ver información detallada sobre la ejecución.
- Estado: el estado de procesamiento de la ejecución del flujo de trabajo. Para obtener más información sobre los estados de ejecución, consulte [Estados de ejecución del flujo de trabajo](#).
- Activador: la persona, la confirmación, la solicitud de extracción (PR) o el programa que inició la ejecución del flujo de trabajo. Para obtener más información, consulte [Trabajar con factores desencadenantes](#).

- Flujo de trabajo: el nombre del flujo de trabajo para el que se inició una ejecución y el repositorio y la rama de origen donde reside el archivo de definición del flujo de trabajo. Puede que tengas que ampliar el ancho de la columna para ver esta información.

 Note

Si esta columna está configurada como No disponible, normalmente se debe a que el flujo de trabajo asociado se ha eliminado o movido.

- Hora de inicio: hora a la que se inició la ejecución del flujo de trabajo.
- Duración: cuánto tiempo tardó en procesarse la ejecución del flujo de trabajo. Las duraciones muy largas o muy cortas pueden indicar problemas.
- Hora de finalización: hora en la que finalizó la ejecución del flujo de trabajo.

Ver el estado y los detalles de todas las ejecuciones de un flujo de trabajo específico

Es posible que desee ver el estado y los detalles de todas las ejecuciones asociadas a un flujo de trabajo específico para ver si alguna ejecución está creando atascos en el flujo de trabajo o para ver qué ejecuciones están actualmente en curso o se han completado.

Para ver el estado y los detalles de todas las ejecuciones de un flujo de trabajo específico

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. En el nombre del flujo de trabajo, selecciona Ejecuciones.

Aparecen las ejecuciones asociadas al flujo de trabajo elegido.

La página está dividida en dos secciones:

- Ejecuciones activas: muestra las ejecuciones en curso. Estas ejecuciones estarán en uno de los siguientes estados: En curso.
- Historial de ejecuciones: muestra las ejecuciones que se han completado (es decir, que no están en curso).

Para obtener más información sobre los estados de las ejecuciones, consulte [Estados de ejecución del flujo de trabajo](#).

Visualización de las ejecuciones de un flujo de trabajo en el diagrama de flujo de trabajo

Puede ver el estado de todas las ejecuciones de un flujo de trabajo a medida que avanzan juntas por el flujo de trabajo. Las ejecuciones se muestran en el diagrama de flujo de trabajo (a diferencia de en una vista de lista). Esto le proporciona una representación visual de qué ejecuciones se están procesando mediante qué acciones y qué ejecuciones están en espera en una cola.

Para ver el estado de varias ejecuciones a medida que avanzan juntas a lo largo de un flujo de trabajo

Note

Este procedimiento solo se aplica si el flujo de trabajo utiliza el modo de ejecución en cola o sustituido. Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#).

1. [Abra la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre del flujo de trabajo que contiene las ejecuciones que desea ver. Puede filtrar por el nombre del repositorio o de la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.

Note

Asegúrese de que está viendo una página de flujo de trabajo y no una página de ejecución.

5. Selecciona la pestaña Estado más reciente en la esquina superior izquierda.

Aparece un diagrama de flujo de trabajo.

6. Revise el diagrama de flujo de trabajo. El diagrama muestra todas las ejecuciones que están actualmente en curso dentro del flujo de trabajo y las últimas ejecuciones que han finalizado. Más específicamente:
- Las carreras que aparecen en la parte superior, antes que las Fuentes, están en cola y esperando para empezar.
 - Las corridas que aparecen entre acciones se ponen en cola y esperan ser procesadas por la siguiente acción.
 - Las ejecuciones que aparecen dentro de una acción son 1. que la acción está procesando actualmente, 2. que la acción ha terminado de procesar o 3. que la acción no las ha procesado (normalmente porque una acción anterior ha fallado).

Configuración de ejecuciones en cola, reemplazadas y paralelas

De forma predeterminada, cuando se ejecutan varios flujos de trabajo al mismo tiempo, los pone en CodeCatalyst cola y los procesa uno por uno, en el orden en que se iniciaron. Puede cambiar este comportamiento predeterminado especificando un modo de ejecución. Hay varios modos de ejecución:

- (Predeterminado) Modo de ejecución en cola: CodeCatalyst los procesos se ejecutan uno por uno
- Modo de ejecución reemplazado: CodeCatalyst los procesos se ejecutan uno por uno, y las ejecuciones más nuevas sustituyen a las antiguas
- Modo de ejecución paralela: CodeCatalyst los procesos se ejecutan en paralelo

Temas

- [Acerca del modo de ejecución en cola](#)
- [Acerca del modo de ejecución reemplazado](#)
- [Acerca del modo de ejecución paralela](#)
- [Cambiar el modo de ejecución](#)

Acerca del modo de ejecución en cola

En el modo de ejecución en cola, las ejecuciones se producen en serie y las ejecuciones en espera forman una cola.

Las colas se forman en los puntos de entrada a las acciones y los grupos de acciones, por lo que puede tener varias colas en el mismo flujo de trabajo (consulte). [Figure 1](#) Cuando una ejecución en cola entra en una acción, la acción se bloquea y no puede entrar ninguna otra ejecución. Cuando la ejecución finaliza y sale de la acción, la acción se desbloquea y está lista para la siguiente ejecución.

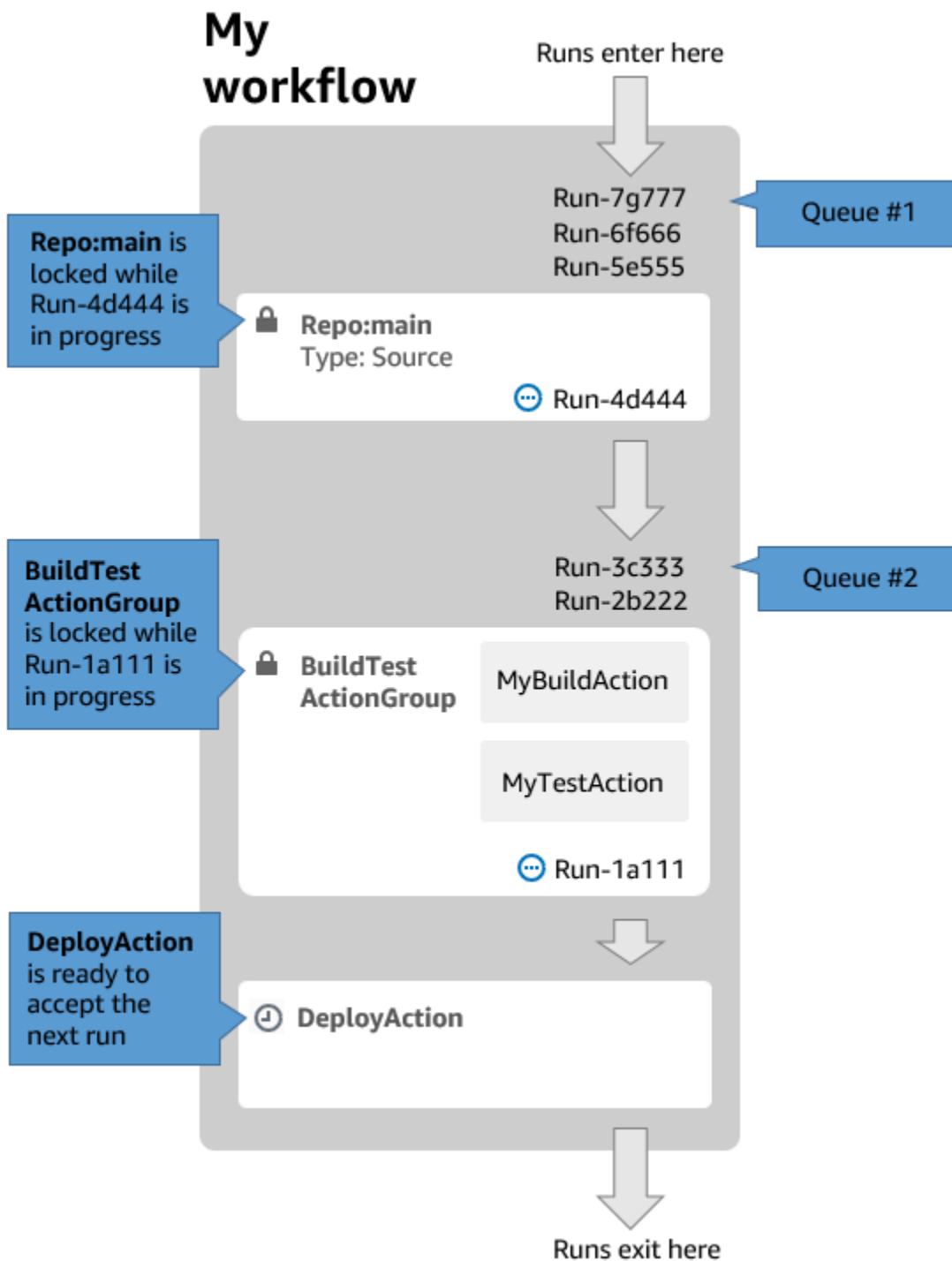
[Figure 1](#) ilustra un flujo de trabajo configurado en modo de ejecución en cola. En él se muestra lo siguiente:

- Siete ejecuciones recorriendo el flujo de trabajo.
- Dos colas: una fuera de la entrada a la fuente de entrada (Repo:main) y otra fuera de la entrada a la acción. BuildTestActionGroup
- Dos bloques bloqueados: la fuente de entrada (Repo:main) y el. BuildTestActionGroup

Así es como se desarrollarán las cosas a medida que se ejecute el flujo de trabajo y finalice el procesamiento:

- Cuando RUN-4D444 termine de clonar el repositorio de fuentes, saldrá de la fuente de entrada y se unirá a la cola detrás de RUN-3C333. A continuación, el RUN-5e555 ingresará a la fuente de entrada.
- Cuando el Run-1a111 termine de construirse y probarse, saldrá de la BuildTestActionGroup acción y entrará en ella. DeployAction Luego, Run-2b222 entrará en acción. BuildTestActionGroup

Figura 1: Un flujo de trabajo configurado en «modo de ejecución en cola»



Utilice el modo de ejecución en cola si:

- Desea mantener una one-to-one relación entre las funciones y las ejecuciones; estas funciones pueden agruparse cuando se utiliza el modo sustituido. Por ejemplo, cuando se fusiona la entidad 1 en la confirmación 1, se inicia la ejecución 1, y cuando se fusiona la entidad 2 en la confirmación

2, se inicia la ejecución 2, y así sucesivamente. Si utilizara el modo sustituido en lugar del modo en cola, sus entidades (y confirmaciones) se agruparían en la ejecución que sustituirá a las demás.

- Quieres evitar las condiciones de carrera y los problemas inesperados que puedan producirse al utilizar el modo paralelo. Por ejemplo, si dos desarrolladores de software, Wang y Saanvi, inician las ejecuciones del flujo de trabajo aproximadamente al mismo tiempo para implementarlas en un clúster de Amazon ECS, la ejecución de Wang podría iniciar las pruebas de integración en el clúster, mientras que la ejecución de Saanvi implementa un nuevo código de aplicación en el clúster, lo que provoca que las pruebas de Wang fallen o prueben el código incorrecto. Como otro ejemplo, es posible que tengas un objetivo que no tenga un mecanismo de bloqueo, en cuyo caso las dos corridas podrían sobrescribir los cambios de la otra de forma inesperada.
- Desea limitar la carga de los recursos informáticos que se CodeCatalyst utilizan para procesar las ejecuciones. Por ejemplo, si tienes tres acciones en tu flujo de trabajo, puedes realizar un máximo de tres ejecuciones al mismo tiempo. Imponer un límite al número de ejecuciones que se pueden realizar a la vez hace que el rendimiento de las ejecuciones sea más predecible.
- Desea limitar el número de solicitudes que el flujo de trabajo realiza a servicios de terceros. Por ejemplo, su flujo de trabajo puede tener una acción de creación que incluya instrucciones para extraer una imagen de Docker Hub. [Docker Hub limita el número de solicitudes de cambios](#) que puedes realizar a un número determinado por hora y por cuenta, y si superas el límite, quedarás bloqueado. Si utilizas el modo de ejecución en cola para reducir el rendimiento de las ejecuciones, se generarán menos solicitudes a Docker Hub por hora, lo que limitará la posibilidad de que se produzcan bloqueos y los consiguientes errores de compilación y ejecución.

Tamaño máximo de cola: 50

Notas sobre el tamaño máximo de la cola:

- El tamaño máximo de la cola se refiere al número máximo de ejecuciones permitido en todas las colas del flujo de trabajo.
- Si una cola supera las 50 corridas, descarta la 51.^a y CodeCatalyst las siguientes.

Comportamiento de error:

Si una ejecución deja de responder mientras está siendo procesada por una acción, las carreras que estén detrás de ella permanecerán en la cola hasta que se agote el tiempo de espera de la acción. Las acciones se agotan al cabo de una hora.

Si una ejecución falla dentro de una acción, se permite continuar con la primera ejecución en cola que haya quedado detrás de ella.

Acerca del modo de ejecución reemplazado

El modo de ejecución sustituida es el mismo que el modo de ejecución en cola, con la salvedad de que:

- Si una ejecución en cola alcanza a otra ejecución de la cola, la ejecución posterior reemplaza (reemplaza) a la anterior, y esta última se cancela y se marca como «reemplazada».
- Como resultado del comportamiento descrito en el primer bullet, una cola solo puede incluir una ejecución cuando se utiliza el modo de ejecución reemplazado.

Si se utiliza el flujo de trabajo [Figure 1](#) como guía, si se aplica el modo de ejecución sustituido a este flujo de trabajo, se produciría lo siguiente:

- El run-7g777 sustituiría a las otras dos corridas de la cola y sería la única ejecución que quedara en la cola #1. La Run-6f666 y la Run-5e555 se cancelarían.
- La Run-3c333 reemplazaría a la Run-2b222 y sería la única carrera que quedara en la cola #2. La run-2b222 se cancelaría.

Utilice el modo de ejecución sustituido si desea:

- mejor rendimiento que con el modo en cola
- incluso menos solicitudes a servicios de terceros que con el modo en cola; esto es ventajoso si el servicio de terceros tiene límites de velocidad, como Docker Hub

Acerca del modo de ejecución paralela

En el modo de ejecución paralela, las ejecuciones son independientes entre sí y no hay que esperar a que se completen las demás para empezar. No hay colas y el rendimiento de las ejecuciones solo está limitado por la rapidez con la que se completen las acciones del flujo de trabajo.

Utilice el modo de ejecución paralela en entornos de desarrollo en los que cada usuario tiene su propia rama de funciones y se despliega en destinos que no comparten otros usuarios.

⚠ Important

Si tiene un objetivo compartido en el que pueden implementarlo varios usuarios, como una función Lambda en un entorno de producción, no utilice el modo paralelo, ya que pueden producirse condiciones de carrera. Se produce una condición de carrera cuando las ejecuciones de un flujo de trabajo en paralelo intentan cambiar un recurso compartido al mismo tiempo, lo que genera resultados impredecibles.

Número máximo de corridos paralelos: 1000 por CodeCatalyst espacio

Cambiar el modo de ejecución

Puede configurar el modo de ejecución en cola, reemplazado o paralelo. El valor predeterminado es en cola.

Al cambiar el modo de ejecución de en cola o reemplazado a paralelo, CodeCatalyst cancela las ejecuciones que están en cola y permite que las ejecuciones que una acción está procesando actualmente finalicen antes de cancelarlas.

Al cambiar el modo de ejecución de paralelo a en cola o reemplazado, CodeCatalyst permite que se completen todas las ejecuciones paralelas que se estén ejecutando actualmente. Todas las ejecuciones que se inicien después de cambiar el modo de ejecución a en cola o sustituido utilizarán el nuevo modo.

Visual

Para cambiar el modo de ejecución mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. En la parte superior derecha, selecciona Propiedades del flujo de trabajo.
7. Expanda Avanzado y, en Modo de ejecución, elija una de las siguientes opciones:
 - a. En cola: consulte [Acerca del modo de ejecución en cola](#)

- b. Reemplazado: consulte [Acerca del modo de ejecución reemplazado](#)
- c. Paralelo: véase [Acerca del modo de ejecución paralela](#)
8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para cambiar el modo de ejecución mediante el editor YAML

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Añada la RunMode propiedad, de la siguiente manera:

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED|SUPERSEDED|PARALLEL
```

Para obtener más información, consulte la descripción de la RunMode propiedad en la [Propiedades de primer nivel](#) sección del [Referencia de definición de flujo de trabajo](#).

8. (Opcional) Elija Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Trabajando con secretos

En ocasiones, es posible que necesites usar datos confidenciales, como las credenciales de autenticación, en tus flujos de trabajo. Debe evitarse almacenar estos valores en texto plano en

cualquier parte del repositorio, ya que cualquier persona que tenga acceso al repositorio que contiene el secreto puede verlos. Del mismo modo, estos valores no deberían usarse directamente en ninguna definición de flujo de trabajo, ya que estarán visibles como archivos en tu repositorio. Con CodeCatalyst, puedes proteger estos valores añadiendo un secreto a tu proyecto y, a continuación, haciendo referencia al secreto en el archivo de definición del flujo de trabajo. Ten en cuenta que puedes tener un máximo de cinco secretos por acción.

Note

Los secretos solo se pueden usar para reemplazar las contraseñas y la información confidencial del archivo de definición del flujo de trabajo.

Temas

- [Crear un secreto](#)
- [Edición de un secreto](#)
- [Uso de un secreto](#)
- [Eliminar un secreto](#)

Crear un secreto

Utilice el siguiente procedimiento para crear un secreto. El secreto contiene la información confidencial que desea ocultar.

Note

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

Para crear un secreto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, Secretos.
3. Elija Create secret (Crear secreto).
4. Introduzca la información siguiente:

Nombre

Introduce un nombre para tu secreto.

Valor

Introduce el valor del secreto. Esta es la información confidencial que quieres ocultar de la vista. De forma predeterminada, el valor no se muestra. Para mostrar el valor, seleccione **Mostrar valor**.

Descripción

(Opcional) Introduce una descripción para tu secreto.

5. Seleccione **Crear**.

Edición de un secreto

Utilice el siguiente procedimiento para editar un secreto.

Para editar un secreto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija **CI/CD** y, a continuación, **Secretos**.
3. En la lista de secretos, elija el secreto que desee editar.
4. Elija **Editar**.
5. Edite las siguientes propiedades:

Valor

Introduzca el valor del secreto. Este es el valor que quieres ocultar de la vista. De forma predeterminada, el valor no se muestra.

Descripción

(Opcional) Introduce una descripción para tu secreto.

6. Seleccione **Guardar**.

Uso de un secreto

Para usar un secreto en una acción de flujo de trabajo, debe obtener el identificador de referencia del secreto y usar ese identificador en la acción del flujo de trabajo.

Temas

- [Obtener el identificador de un secreto](#)
- [Hacer referencia a un secreto en un flujo de trabajo](#)

Obtener el identificador de un secreto

Utilice el siguiente procedimiento para obtener el identificador de referencia del secreto. Añadirá este identificador a su flujo de trabajo.

Para obtener el identificador de referencia del secreto

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, Secretos.
3. En la lista de secretos, busca el secreto que quieres usar.
4. En la columna ID de referencia, copia el identificador del secreto. La siguiente es la sintaxis del ID de referencia:

```
`${Secrets.<name>}`
```

Hacer referencia a un secreto en un flujo de trabajo

Utilice el siguiente procedimiento para hacer referencia a un secreto en un flujo de trabajo.

Para hacer referencia a un secreto

1. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
2. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
3. Elija Editar.
4. Elige YAML.

5. Modifique el YAML para usar el identificador del secreto. Por ejemplo, para usar un nombre de usuario y una contraseña que se almacenan como secretos con el `curl` comando, usaría un Run comando similar al siguiente:

```
- Run: curl -u <username-secret-identifier>:<password-secret-identifier> https://example.com
```

6. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
7. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Eliminar un secreto

Utilice el siguiente procedimiento para eliminar un secreto y el identificador de referencia secreto.

Note

Antes de eliminar un secreto, se recomienda eliminar el identificador de referencia del secreto de todas las acciones del flujo de trabajo. Si elimina el secreto sin eliminar el identificador de referencia, la acción fallará la próxima vez que se ejecute.

Para eliminar el identificador de referencia de un secreto de un flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
3. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
4. Elija Editar.
5. Elige YAML.
6. Busca la siguiente cadena en el flujo de trabajo:

```
${Secrets.
```

Esto busca todos los identificadores de referencia de todos los secretos.

7. Elimine el identificador de referencia del secreto elegido o sustitúyalo por un valor de texto simple.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Para eliminar un secreto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, elija CI/CD y, a continuación, Secretos.
3. En la lista de secretos, elija el secreto que desee eliminar.
4. Elija Eliminar.
5. **delete**Ingresar para confirmar la eliminación.
6. Elija Eliminar.

Trabajar con fuentes

Una fuente, también denominada fuente de entrada, es un repositorio de fuentes al que necesita acceder una [acción de flujo](#) de trabajo para llevar a cabo sus tareas. Por ejemplo, es posible que una acción de flujo de trabajo necesite acceder a una fuente para obtener pruebas unitarias y ejecutarlas en los archivos fuente de la aplicación.

CodeCatalyst los flujos de trabajo admiten las siguientes fuentes:

- CodeCatalyst repositorios de fuentes: para obtener más información, consulte [Repositorios de fuentes en CodeCatalyst](#).
- GitHub repositorios de fuentes: para obtener más información, consulte [Uso de GitHub repositorios en CodeCatalyst](#)

Temas

- [Especificar la fuente que almacenará el archivo de definición del flujo de trabajo](#)
- [Especificar la fuente que utilizará una acción de flujo de trabajo](#)
- [Hacer referencia a archivos de un repositorio fuente](#)
- [Variables generadas por la fuente](#)

Especificar la fuente que almacenará el archivo de definición del flujo de trabajo

Siga las instrucciones siguientes para especificar el repositorio de CodeCatalyst origen en el que desea almacenar el archivo de definición de flujo de trabajo. Si prefiere especificar un repositorio de GitHub origen, consulte en su lugar [Uso de GitHub repositorios en CodeCatalyst](#).

El repositorio de origen en el que reside el archivo de definición de flujo de trabajo se identifica mediante la etiqueta, `WorkflowSource`.

Note

Usted especifica el repositorio de origen en el que reside el archivo de definición de flujo de trabajo cuando lo confirma por primera vez. Tras esta confirmación, el repositorio y el archivo de definición del flujo de trabajo se vinculan permanentemente. La única forma de cambiar el repositorio después de la confirmación inicial es volver a crear el flujo de trabajo en un repositorio diferente.

Para especificar el repositorio de origen que almacenará el archivo de definición del flujo de trabajo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Seleccione Crear flujo de trabajo y cree el flujo de trabajo. Para obtener más información, consulte [Para crear un flujo de trabajo mediante el editor visual](#).

Durante el proceso de creación del flujo de trabajo, se le pide que especifique el CodeCatalyst repositorio en el que desea almacenar el archivo de definición del flujo de trabajo.

Especificar la fuente que utilizará una acción de flujo de trabajo

Usa las siguientes instrucciones para especificar un repositorio de origen para usarlo con una acción de flujo de trabajo. Al iniciarse, la acción agrupa los archivos del repositorio de origen configurado en un artefacto, descarga el artefacto a la [imagen de Docker del entorno de ejecución](#) en la que se ejecuta la acción y, a continuación, completa su procesamiento con los archivos descargados.

Note

Actualmente, dentro de una acción de flujo de trabajo, solo puedes especificar un repositorio de origen, que es el repositorio de origen en el que reside el archivo de definición del flujo de trabajo (en el directorio). `.codecatalyst/workflows/` Este repositorio de origen se representa mediante la etiqueta `WorkflowSource`.

Visual

Para especificar el repositorio de origen que utilizará una acción (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción en la que desee especificar la fuente.
8. Elija Entradas.
9. En Fuentes (opcional), haga lo siguiente:

Especifique las etiquetas que representan los repositorios de origen que necesitará la acción. Actualmente, la única etiqueta admitida es `WorkflowSource` la que representa el repositorio de origen en el que se almacena el archivo de definición de flujo de trabajo.

Si omite una fuente, debe especificar al menos un artefacto de entrada en ella. `action-name/Inputs/Artifacts`

Para obtener más información sobre fuentes, consulte [Trabajar con fuentes](#).

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para especificar el repositorio de origen que utilizará una acción (editor YAML)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una acción, añada un código similar al siguiente:

```
action-name:  
  Inputs:  
    Sources:  
      - WorkflowSource
```

Para obtener más información, consulte la descripción de la `Sources` propiedad en la que [Referencia de definición de flujo de trabajo](#) desea realizar la acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Hacer referencia a archivos de un repositorio fuente

Si tiene archivos que residen en un repositorio de origen y necesita hacer referencia a estos archivos en una de sus acciones de flujo de trabajo, complete el siguiente procedimiento.

Note

Véase también [Hacer referencia a los archivos de un artefacto](#).

Para hacer referencia a un archivo de un repositorio de origen

- En la acción en la que desee hacer referencia a un archivo, añada un código similar al siguiente:

```
Actions:
  My-action:
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - run: cd my-app && cat file1.jar
```

En el código anterior, la acción busca en el `my-app` directorio de la raíz del repositorio `WorkflowSource` fuente para buscar y mostrar el `file1.jar` archivo.

Variables generadas por la fuente

Cuando se ejecuta un flujo de trabajo, su origen produce variables que puede utilizar en acciones posteriores del flujo de trabajo. Para obtener más información, consulte [variables BranchName «» y CommitId «»](#) la [Lista de variables predefinidas](#).

Trabajar con factores desencadenantes

Un activador de flujo de trabajo, o simplemente un activador, te permite iniciar la ejecución automática de un flujo de trabajo cuando se producen determinados eventos, como cuando se introduce un código. Es posible que desee configurar los activadores para evitar que los desarrolladores de software tengan que iniciar manualmente las ejecuciones del flujo de trabajo a través de la CodeCatalyst consola.

Puede utilizar tres tipos de activadores:

- Empujar: un activador de inserción de código hace que se inicie la ejecución de un flujo de trabajo cada vez que se ejecuta una confirmación.
- Solicitud de extracción: cuando se activa una solicitud de extracción, se inicia la ejecución de un flujo de trabajo cada vez que se crea, revisa o cierra una solicitud de extracción.
- Programación: un activador de programación hace que la ejecución de un flujo de trabajo comience según el cronograma que tú definas. Considere la posibilidad de utilizar un activador

programado para ejecutar versiones nocturnas de su software, de modo que la última versión esté lista para que los desarrolladores de software puedan trabajar en ella a la mañana siguiente.

Puedes usar activadores push, pull request y programables solos o combinados en el mismo flujo de trabajo.

Tip

Para ver un desencadenante en acción, lanza un proyecto con un plano. La mayoría de los blueprints contienen un flujo de trabajo con un activador. Busque la `Trigger` propiedad en el archivo de definición del flujo de trabajo del blueprint. Para obtener más información acerca de los esquemas, consulte [Crear un proyecto con un plano](#).

Temas

- [Una configuración de disparadores común](#)
- [Consideraciones sobre los factores desencadenantes al ramificar](#)
- [Añadir un gatillo para pulsar, tirar o programar](#)
- [Ejemplos de desencadenantes](#)

Una configuración de disparadores común

En esta sección se describe cómo configurar los desencadenantes para una estrategia común de lanzamiento y ramificación de software.

Estrategia de lanzamiento y ramificación del software:

- Tiene el código de la aplicación en un repositorio fuente.
- Su `main` rama contiene el código finalizado que siempre está listo para su publicación.
- Los desarrolladores de software realizan los cambios en las ramas de funciones ajenas a la rama `main`.
- Tus desarrolladores de software [crean una solicitud](#) de cambios en la que se les pide que fusionen su rama de funciones con la rama de funciones `main` cuando la función esté lista.

Quieres que esta solicitud de extracción inicie automáticamente un flujo de trabajo que cree y pruebe (pero no despliegue) la aplicación utilizando los archivos de la rama de funciones del desarrollador de software.

- Los desarrolladores de software comprueban la compilación y las pruebas para asegurarse de que todo se ve bien. Luego, [fusionan la solicitud de extracción](#) en la main sucursal.

Desea que la fusión inicie automáticamente un flujo de trabajo que cree e implemente el código de su aplicación.

Configuración propuesta de flujo de trabajo o activación:

Dada la estrategia de ramificación del software descrita anteriormente, es posible que desee utilizar dos flujos de trabajo:

- El flujo de trabajo 1 crea y prueba tu aplicación cuando se crea o revisa una solicitud de extracción.
- El flujo de trabajo 2 crea e implementa tu aplicación cuando se fusiona una solicitud de extracción.

El flujo de trabajo 1 tendría este aspecto:

```
Triggers:
  - Type: PULLREQUEST
  Branches:
    - main
  Events:
    - OPEN
    - REVISION
Actions:
  BuildAction:
    instructions-for-building-the-app
  TestAction:
    instructions-for-test-the-app
```

El código de activación anterior inicia automáticamente la ejecución de un flujo de trabajo cada vez que un desarrollador de software crea una solicitud de extracción (o [modifica una](#)) solicitando la fusión de su rama de funciones con la main rama. CodeCatalyst inicia la ejecución de un flujo de trabajo con el código de la rama fuente (es decir, la rama de funciones del desarrollador). El flujo de trabajo crea e implementa la aplicación.

El flujo de trabajo 2 tendría este aspecto:

```
Triggers:
- Type: PUSH
  Branches:
    - main
Actions:
BuildAction:
  instructions-for-building-the-app
DeployAction:
  instructions-for-deploying-the-app
```

En el código de activación anterior, cuando `main` se produce una fusión, se activa el PUSH activador. CodeCatalyst inicia la ejecución de un flujo de trabajo con el código de la `main` rama (que ahora incluye el código de la solicitud de extracción). El flujo de trabajo crea e implementa la aplicación.

Para obtener instrucciones sobre cómo añadir activadores a un archivo de definición de flujo de trabajo, consulte [Añadir un gatillo para pulsar, tirar o programar](#).

Para ver más ejemplos de desencadenantes y explicaciones adicionales, consulte [Ejemplos de desencadenantes](#).

Consideraciones sobre los factores desencadenantes al ramificar

En esta sección, se describen algunas de las principales consideraciones a la hora de configurar activadores que incluyan sucursales.

- Consideración 1: Tanto para los activadores de solicitudes push como para los de extracción, si va a especificar una rama, debe especificar la rama de destino (o «destino») en la configuración del disparador. Nunca especifique la rama de origen (o «de»).

En el siguiente ejemplo, una pulsación desde cualquier rama para `main` activar el flujo de trabajo.

```
Triggers:
- Type: PUSH
  Branches:
    - main
```

En el siguiente ejemplo, una solicitud de extracción desde cualquier sucursal `main` activa el flujo de trabajo.

Triggers:

- Type: PULLREQUEST

Branches:

- main

Events:

- OPEN
- REVISION

- Consideración 2: En el caso de los activadores push, una vez activado el flujo de trabajo, el flujo de trabajo se ejecutará con el archivo de definición del flujo de trabajo y los archivos fuente de la rama de destino.
- Consideración 3: En el caso de los activadores de solicitudes de extracción, una vez activado el flujo de trabajo, el flujo de trabajo se ejecutará con el archivo de definición del flujo de trabajo y los archivos fuente de la rama de origen (aunque hayas especificado la rama de destino en la configuración del desencadenador).
- Consideración 4: Es posible que el mismo activador exacto en una rama no se ejecute en otra rama.

Considera el siguiente disparador pulsador:

Triggers:

- Type: PUSH

Branches:

- main

Si el archivo de definición del flujo de trabajo que contiene este activador existe `main` y se clona en `test`, el flujo de trabajo nunca empezará automáticamente a utilizar los archivos incluidos `test` (aunque puede iniciar el flujo de trabajo manualmente si utiliza los archivos `test`).

Revise las consideraciones 1 y 2 para entender por qué el flujo de trabajo nunca se ejecutará automáticamente con los archivos incluidos. `test`

Considera también el siguiente desencadenante de una solicitud de atracción:

Triggers:

- Type: PULLREQUEST

Branches:

- main

Events:

- OPEN

- REVISION

Si el archivo de definición del flujo de trabajo que contiene este activador existe en `main`, el flujo de trabajo nunca se ejecutará con los archivos que contienen `main`. (Sin embargo, si crea una `test` ramificación a partir de `main`, el flujo de trabajo se ejecutará con los archivos incluidos) `test`. Revise las consideraciones 1 y 3 para entender por qué.

Añadir un gatillo para pulsar, tirar o programar

Sigue las siguientes instrucciones para añadir un activador de inserción, tracción o programación a tu flujo de trabajo.

Visual

Para añadir un disparador (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, seleccione las casillas Origen y Desencadenadores.
8. En el panel de configuración, seleccione Añadir disparador.
9. En el cuadro de diálogo Agregar activador, introduzca la información en los campos, de la siguiente manera.

Tipo de disparador

Especifique el tipo de disparador. Puede utilizar uno de los siguientes valores:

- Push (editor visual) o PUSH (editor YAML)

Un disparador push inicia la ejecución de un flujo de trabajo cuando se envía un cambio a tu repositorio de origen. La ejecución del flujo de trabajo utilizará los archivos de la rama a la que vayas a realizar el envío (es decir, la rama de destino).

- Pull Request (editor visual) o PULLREQUEST (editor YAML)

Un activador de solicitudes de extracción inicia la ejecución de un flujo de trabajo cuando se abre, actualiza o cierra una solicitud de extracción en tu repositorio de origen. La ejecución del flujo de trabajo utilizará los archivos de la rama de la que estás extrayendo (es decir, la rama de origen).

- Programación (editor visual) o SCHEDULE (editor YAML)

Un activador de programación inicia las ejecuciones del flujo de trabajo según una programación definida por una expresión cron que usted especifique. Se iniciará una ejecución de flujo de trabajo independiente para cada rama del repositorio de origen utilizando los archivos de la rama. (Para limitar las ramas en las que se activa el activador, usa el campo Ramas (editor visual) o la Branches propiedad (editor YAML).)

Al configurar un activador programado, siga estas pautas:

- Utilice solo un activador de programación por flujo de trabajo.
- Si ha definido varios flujos de trabajo en su CodeCatalyst espacio, le recomendamos que no programe más de 10 de ellos para que se inicien simultáneamente.
- Asegúrese de configurar la expresión cron del disparador con el tiempo suficiente entre ejecuciones. Para obtener más información, consulte [Expression](#).

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

Eventos para la solicitud de cambios

Este campo solo aparece si seleccionaste el tipo de activación de la solicitud de extracción.

Especifica el tipo de eventos de solicitud de extracción que iniciarán la ejecución de un flujo de trabajo. Los valores válidos son los siguientes:

- Se crea una solicitud de extracción (editor visual) o OPEN (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se crea una solicitud de extracción.

- La solicitud de extracción está cerrada (editor visual) o CLOSED (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se cierra una solicitud de extracción. El comportamiento del CLOSED evento es complicado y se entiende mejor con un ejemplo.

Para obtener más información, consulte [Ejemplo: un gatillo al que se tira, se ramifica y se activa el evento «CERRADO»](#).

- Se ha realizado una nueva revisión en pull request (editor visual) o REVISION (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se crea una revisión de una solicitud de extracción. La primera revisión se crea cuando se crea la solicitud de extracción. Después de eso, se crea una nueva revisión cada vez que alguien envía una nueva confirmación a la rama de origen especificada en la solicitud de extracción. Si incluyes el REVISION evento en el activador de la solicitud de cambios, puedes omitirlo, ya que REVISION es un superconjunto de. OPEN OPEN

Puedes especificar varios eventos en el mismo activador de la solicitud de atracción.

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

Programación

Este campo solo aparece si seleccionaste el tipo de activación Programar.

Especifique la expresión cron que describe cuándo desea que se ejecute el flujo de trabajo programado.

Las expresiones cron utilizadas CodeCatalyst utilizan la siguiente sintaxis de seis campos, en la que cada campo está separado por un espacio:

minutos , horas days-of-month, mes, año days-of-week

Ejemplos de expresiones cron

Minutos	Horas	Días del mes	Mes	Días de la semana	Año	Significado
0	0	?	*	MON-FRI	*	Ejecuta un flujo de trabajo a medianoche (UTC +0) de lunes a viernes.
0	2	*	*	?	*	Ejecuta un flujo de trabajo a las 2:00 a. m. (UTC +0) todos los días.
15	22	*	*	?	*	Ejecuta un flujo de trabajo a las 22:15 (UTC+0) todos los días.

Minutos	Horas	Días del mes	Mes	Días de la semana	Año	Significado
0/30	22-2	?	*	SAT-SUN	*	Ejecuta un flujo de trabajo cada 30 minutos de sábado a domingo, entre las 22:00 del día de inicio y las 2:00 del día siguiente (UTC+0).
45	13	L	*	?	2023-2027	Ejecuta un flujo de trabajo a las 13:45 (UTC+0) del último día del mes entre los años 2023 y 2027, ambos inclusive.

Al especificar las expresiones cron CodeCatalyst, asegúrate de seguir estas pautas:

- Especifique una sola expresión cron por SCHEDULE activador.
- Escribe la expresión cron entre comillas dobles (") en el editor YAML.
- Especifique la hora en hora universal coordinada (UTC). No se admiten otras zonas horarias.
- Configure al menos 30 minutos entre ejecuciones. No se admite una cadencia más rápida.
- Especifique el *days-of-week* campo *days-of-month*, pero no ambos. Si especifica un valor o un asterisco (*) en uno de los campos, debe utilizar un signo de interrogación (?) en el otro. El asterisco significa «todos» y el signo de interrogación significa «cualquiera».

Para obtener más ejemplos de expresiones cron e información sobre caracteres comodín?, como, y *L, consulte la [referencia sobre expresiones cron en](#) la Guía del usuario de Amazon EventBridge . Las expresiones cron CodeCatalyst funcionan EventBridge y funcionan exactamente de la misma manera.

Para ver ejemplos de activadores de horarios, consulte [Ejemplos de desencadenantes](#).

Ramas y patrón de ramas

(Opcional)

Especifique las ramas del repositorio de origen que el activador supervisa para saber cuándo iniciar la ejecución de un flujo de trabajo. Puedes usar patrones de expresiones regulares para definir los nombres de tus ramas. Por ejemplo, úsalo `main.*` para hacer coincidir todas las ramas que comiencen por `main`

Las ramas que se deben especificar son diferentes en función del tipo de disparador:

- En el caso de un disparador automático, especifique las ramas a las que va a enviar el mensaje, es decir, las sucursales de destino. Se iniciará una ejecución de flujo de trabajo por cada sucursal coincidente, utilizando los archivos de la sucursal coincidente.

Ejemplos: `main.*`, `mainline`

- Para activar una solicitud de cambios, especifica las sucursales a las que vas a enviar mensajes, es decir, las sucursales de destino. Se iniciará una ejecución de flujo de trabajo por cada rama coincidente, utilizando el archivo de definición del flujo de trabajo y los archivos fuente de la rama de origen (no de la rama coincidente).

Ejemplos: `main.*`, `mainline`, `v1\-.*` (coincide con las ramas que comienzan con `v1-`)

- Para activar una programación, especifique las ramas que contienen los archivos que desea que utilice la ejecución programada. Se iniciará una ejecución de flujo de trabajo por cada rama coincidente, utilizando el archivo de definición del flujo de trabajo y los archivos fuente de la rama coincidente.

Ejemplos: `main.*`, `version\-1\.*`

Note

Si no especificas las ramas, el activador monitoriza todas las ramas del repositorio de origen e iniciará una ejecución de flujo de trabajo con el archivo de definición del flujo de trabajo y los archivos fuente de:

- La rama hacia la que estás presionando (para los activadores de pulsación). Para obtener más información, consulte [Ejemplo: un sencillo disparador por pulsación de código](#).
- La sucursal desde la que realizas la extracción (para activar las solicitudes de extracción). Para obtener más información, consulte [Ejemplo: un sencillo activador de una solicitud de extracción](#).
- Todas las sucursales (para programar activaciones). Se iniciará una ejecución de flujo de trabajo por cada rama del repositorio de origen. Para obtener más información, consulte [Ejemplo: un sencillo disparador programado](#).

Para obtener más información sobre las ramas y los activadores, consulte [Consideraciones sobre los factores desencadenantes al ramificar](#).

Para obtener más ejemplos, consulte [Ejemplos de desencadenantes](#).

Se han modificado los archivos

Este campo solo aparece si ha seleccionado el tipo de activación de la solicitud Push o Pull.

Especifique los archivos o carpetas del repositorio de origen que el activador supervisa para saber cuándo iniciar la ejecución de un flujo de trabajo. Puede utilizar expresiones regulares para hacer coincidir los nombres o las rutas de los archivos.

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para añadir un disparador (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. Añada una Triggers sección y propiedades subyacentes utilizando el siguiente ejemplo como guía. Para obtener más información, consulte [Triggers](#) en la [Referencia de definición de flujo de trabajo](#).

Un disparador de inserción de código podría tener este aspecto:

```
Triggers:  
  - Type: PUSH  
    Branches:  
      - main
```

Un activador de pull request podría tener este aspecto:

```
Triggers:  
  - Type: PULLREQUEST  
    Branches:  
      - main.*  
    Events:  
      - OPEN
```

- REVISION
- CLOSED

Un activador programado podría tener este aspecto:

Triggers:

- Type: SCHEDULE

Branches:

- main.*

Run the workflow at 1:15 am (UTC+0) every Friday until the end of 2023

Expression: "15 1 ? * FRI 2022-2023"

Para ver más ejemplos de expresiones cron que puede utilizar en la Expression propiedad, consulte [Expression](#).

Para ver más ejemplos de activadores push, pull request y schedule, consulte [Ejemplos de desencadenantes](#).

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Ejemplos de desencadenantes

Los siguientes ejemplos muestran cómo añadir distintos tipos de activadores en el archivo de definición del flujo de trabajo.

Temas

- [Ejemplo: un sencillo disparador por pulsación de código](#)
- [Ejemplo: un simple disparador de «pulsar el botón principal»](#)
- [Ejemplo: un sencillo activador de una solicitud de extracción](#)
- [Ejemplo: un sencillo disparador programado](#)
- [Ejemplo: un disparador con un horario y ramificaciones](#)
- [Ejemplo: un disparador con un horario, una pulsación y ramificaciones](#)
- [Ejemplo: un gatillo que se tira y se ramifica](#)
- [Ejemplo: un gatillo al que se tira, se ramifica y se activa el evento «CERRADO»](#)

- [Ejemplo: un disparador con una pulsación, ramas y archivos](#)

Ejemplo: un sencillo disparador por pulsación de código

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se envía código a cualquier rama del repositorio de código fuente.


Cuando se activa este activador, se CodeCatalyst inicia una ejecución de flujo de trabajo con los archivos de la rama a la que estás accediendo (es decir, la rama de destino).

Por ejemplo, si presionas una confirmación `main`, se CodeCatalyst inicia una ejecución de flujo de trabajo utilizando el archivo de definición del flujo de trabajo y otros archivos fuente. `main`

Como otro ejemplo, si se envía una confirmación a `feature-branch-123`, se CodeCatalyst inicia una ejecución de flujo de trabajo utilizando el archivo de definición del flujo de trabajo y otros archivos fuente activados. `feature-branch-123`

Triggers:

- Type: PUSH

 Note

Si desea que la ejecución de un flujo de trabajo se inicie solo cuando presiones `main`, consulte. [Ejemplo: un simple disparador de «pulsar el botón principal»](#)

Ejemplo: un simple disparador de «pulsar el botón principal»

En el siguiente ejemplo, se muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se envía código a la rama (`main` y solo a la rama) del `main` repositorio de código fuente.

Triggers:

- Type: PUSH

Branches:

- `main`

Ejemplo: un sencillo activador de una solicitud de extracción

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se crea o revisa una solicitud de extracción en tu repositorio de origen.

Cuando se activa este activador, CodeCatalyst inicia la ejecución de un flujo de trabajo con el archivo de definición del flujo de trabajo y otros archivos fuente de la rama de la que estás extrayendo (es decir, la rama de origen).

Por ejemplo, si creas una solicitud de extracción con una rama de origen llamada `feature-123` y una rama de destino llamada `main`, CodeCatalyst inicia una ejecución de flujo de trabajo utilizando el archivo de definición del flujo de trabajo y otros archivos de origen. `feature-123`

Triggers:

- Type: PULLREQUEST

Events:

- OPEN
- REVISION

Ejemplo: un sencillo disparador programado

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo a medianoche (UTC+0) de lunes a viernes.

Cuando se activa este activador, se CodeCatalyst inicia una sola ejecución de flujo de trabajo para cada rama del repositorio de origen que contenga un archivo de definición de flujo de trabajo con este activador.

Por ejemplo, si tienes tres ramas en tu repositorio de origen, `main` `release-v1` `feature-123`, y cada una de estas ramas contiene un archivo de definición de flujo de trabajo con el siguiente desencadenante, CodeCatalyst inicia tres ejecuciones de flujo de trabajo: una con los archivos de `main`, otra con los archivos de `release-v1` y otra con los archivos de `feature-123`.

Triggers:

- Type: SCHEDULE

Expression: "`0 0 ? * MON-FRI *`"

Para ver más ejemplos de expresiones cron que puede utilizar en la `Expression` propiedad, consulte [Expression](#).

Ejemplo: un disparador con un horario y ramificaciones

El siguiente ejemplo muestra un disparador que inicia una ejecución de flujo de trabajo a las 18:15 (UTC+0) todos los días.

Cuando se activa este activador, CodeCatalyst inicia una ejecución de flujo de trabajo con los archivos de la `main` rama e inicia ejecuciones adicionales para cada rama que comience con `release-`.

Por ejemplo, si tiene ramas denominadas `main`, `release-v1bugfix-1`, y `bugfix-2` en el repositorio de origen, CodeCatalyst inicia dos ejecuciones de flujo de trabajo: una con los archivos de `y` otra con los archivos de `release-v1`. `main` No inicia las ejecuciones de flujo de trabajo de las `bugfix-1` ramas `bugfix-1` y.

Triggers:

- Type: SCHEDULE
- Expression: "15 18 * * ? *"
- Branches:
 - main
 - release\-.*

Para ver más ejemplos de expresiones cron que puede utilizar en la `Expression` propiedad, consulte [Expression](#).

Ejemplo: un disparador con un horario, una pulsación y ramificaciones

El siguiente ejemplo muestra un activador que inicia un flujo de trabajo que se ejecuta a medianoche (UTC+0) todos los días y siempre que se envía código a la sucursal. `main`

En este ejemplo:

- La ejecución de un flujo de trabajo comienza todos los días a medianoche. La ejecución del flujo de trabajo utiliza el archivo de definición del flujo de trabajo y otros archivos fuente de la `main` rama.
- La ejecución de un flujo de trabajo también se inicia cada vez que envías una confirmación a la `main` rama. La ejecución del flujo de trabajo utiliza el archivo de definición del flujo de trabajo y otros archivos de origen de la rama de destino (`main`).

Triggers:

- Type: SCHEDULE
- Expression: "0 0 * * ? *"
- Branches:
 - main
- Type: PUSH

```
Branches:
```

```
- main
```

Para ver más ejemplos de expresiones cron que puede utilizar en la `Expression` propiedad, consulte [Expression](#).

Ejemplo: un gatillo que se tira y se ramifica

En el siguiente ejemplo, se muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que alguien abre o modifica una solicitud de extracción con una sucursal de destino llamada `main`. Aunque la rama especificada en la `Triggers` configuración lo sea `main`, la ejecución del flujo de trabajo utilizará el archivo de definición del flujo de trabajo y otros archivos fuente de la rama de origen (que es la rama de la que se extrae).

```
Triggers:
```

```
- Type: PULLREQUEST
```

```
Branches:
```

```
- main
```

```
Events:
```

```
- OPEN
```

```
- REVISION
```

Ejemplo: un gatillo al que se tira, se ramifica y se activa el evento «CERRADO»

En el siguiente ejemplo, se muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se cierra una solicitud de extracción en una rama que comienza con `main`.

En este ejemplo:

- Cuando cierras una solicitud de extracción con una rama de destino por la que comienza `main`, una ejecución de flujo de trabajo se inicia automáticamente con el archivo de definición del flujo de trabajo y otros archivos fuente de la rama de origen (ahora cerrada).
- Si has configurado tu repositorio de origen para que elimine las ramas automáticamente después de fusionar una solicitud de extracción, estas sucursales nunca tendrán la oportunidad de entrar en ese `CLOSED` estado. Esto significa que las sucursales fusionadas no activarán el `CLOSED` activador de la solicitud de extracción. La única forma de activar el `CLOSED` activador en este escenario es cerrar la solicitud de extracción sin fusionarla.

```
Triggers:
```



```
- Type: PULLREQUEST
Branches:
  - main.*
Events:
  - CLOSED
```

Ejemplo: un disparador con una pulsación, ramas y archivos

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se realiza un cambio en el `filename.txt` archivo, o en cualquier archivo del `src` directorio, de la `main` rama.

Cuando se activa este activador, CodeCatalyst inicia una ejecución de flujo de trabajo con el archivo de definición del flujo de trabajo y otros archivos fuente de la `main` rama.

```
Triggers:
- Type: PUSH
Branches:
  - main
FilesChanged:
  - filename.txt
  - src\*.*
```

Trabajar con variables

Una variable es un par clave-valor que contiene información a la que puede hacer referencia en su CodeCatalyst flujo de trabajo.

Hay dos tipos de variables que puede usar en un flujo de trabajo:

- Variables definidas por el usuario: son pares clave-valor que usted define.
- Variables predefinidas: son pares clave-valor que un flujo de trabajo emite automáticamente. No es necesario que los defina.

Note

CodeCatalyst también admite [parámetros GitHub de salida](#), que se comportan como variables y se puede hacer referencia a ellos en otras acciones. Para obtener más información, consulte [Trabajar con los parámetros de salida de GitHub Actions](#).

Temas

- [Uso de variables definidas por el usuario](#)
- [Uso de variables predefinidas](#)

Uso de variables definidas por el usuario

Las variables definidas por el usuario son pares clave-valor que usted define. Existen dos tipos:

- Variables de texto simple o simplemente variables: son pares clave-valor que se definen en texto sin formato en el archivo de definición del flujo de trabajo.
- Secretos: son pares clave-valor que se definen en una página de secretos independiente de la consola de Amazon CodeCatalyst . La clave (nombre) es una etiqueta pública y el valor contiene la información que deseas mantener privada. Solo debe especificar la clave en el archivo de definición del flujo de trabajo. Utilice secretos en lugar de contraseñas y otra información confidencial en el archivo de definición del flujo de trabajo.

Note

Por motivos de brevedad, en esta guía se utiliza el término `variable` para referirse a una variable de texto sin formato.


Temas

- [Definir una variable](#)
- [Definir un secreto](#)
- [Exportar una variable para que otras acciones puedan utilizarla](#)
- [Hacer referencia a una variable en la acción que la define](#)
- [Hacer referencia a una salida variable mediante otra acción](#)
- [Hacer referencia a un secreto](#)
- [Ejemplos](#)

Definir una variable

Puede definir las variables de dos maneras:

- En la Inputs sección de una acción de flujo de trabajo, consulte [Para definir una variable en la sección «Entradas»](#)
- En la Steps sección de una acción de flujo de trabajo, consulte [Para definir una variable en la sección «Pasos»](#)

 Note

El Steps método solo funciona con las acciones de CodeCatalyst compilación, prueba y GitHub acciones, ya que son las únicas acciones que incluyen una Steps sección.

Para ver ejemplos, consulte [Ejemplos](#).

Visual

Para definir una variable en la sección «Entradas» (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción en la que desee establecer la variable.
8. Elija Entradas.
9. En Variables (opcional), elija Añadir variable y, a continuación, haga lo siguiente:

Especifique una secuencia de pares de nombre/valor que defina las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para definir una variable en la sección «Entradas» (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una acción de flujo de trabajo, añada un código similar al siguiente:

```
action-name:  
  Inputs:  
    Variables:  
      - Name: variable-name  
        Value: variable-value
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener más información, consulta la [Referencia de definición de flujo de trabajo](#) sección para tu acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Visual

Para definir una variable en la sección «Pasos» (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.

2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción en la que desee establecer la variable.
8. Elija Configuration (Configuración).
9. En los comandos de Shell o en GitHublas acciones (YAML), las que estén disponibles, defina una variable en la acciónSteps, de forma explícita o implícita.
 - Para definir la variable de forma explícita, inclúyela en un comando bash directamente en la sección. Steps
 - Para definir una variable de forma implícita, especifícala en un archivo al que se haga referencia en la sección de Steps la acción.

Para ver ejemplos, consulte [Ejemplos](#). Para obtener más información, consulta la sección [Referencia de definición de flujo de trabajo](#) correspondiente a la acción.

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para definir una variable en la sección «Pasos» (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.

7. En una acción de flujo de trabajo, defina una variable en la Steps sección de la acción, de forma explícita o implícita.
 - Para definir la variable de forma explícita, inclúyela en un comando bash directamente en la Steps sección.
 - Para definir una variable de forma implícita, especifícala en un archivo al que se haga referencia en la sección de Steps la acción.

Para ver ejemplos, consulte [Ejemplos](#). Para obtener más información, consulta la sección [Referencia de definición de flujo de trabajo](#) correspondiente a la acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Definir un secreto

El secreto se define en la página de secretos de la CodeCatalyst consola. Para obtener más información, consulte [Trabajando con secretos](#).

Por ejemplo, puedes definir un secreto con el siguiente aspecto:

- Nombre (clave): **my-password**
- Valor: **^*H3#!b9**

Una vez definido el secreto, puede especificar la clave del secreto (**my-password**) en el archivo de definición del flujo de trabajo. Para ver un ejemplo práctico, consulte [Ejemplo: hacer referencia a un secreto](#).

Exportar una variable para que otras acciones puedan utilizarla

Siga las instrucciones siguientes para exportar una variable de una acción para poder hacer referencia a ella en otras acciones.

Antes de exportar una variable, tenga en cuenta lo siguiente:

- Si solo necesitas hacer referencia a la variable dentro de la acción en la que está definida, no necesitas exportarla.

- No todas las acciones admiten la exportación de variables. Para determinar si la acción admite esta función, siga las instrucciones del editor visual que aparecen a continuación y compruebe si la acción incluye un botón de variables en la pestaña Salidas. En caso afirmativo, se admite la exportación de variables.
- Para exportar una variable de una GitHub acción, consulte [Exportación GitHub de un parámetro de salida para que otras acciones puedan usarlo](#).

Requisito previo

Asegúrese de haber definido la variable que desea exportar. Para obtener más información, consulte [Definir una variable](#).

Visual

Para exportar una variable (editor visual)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elija la acción desde la que desee exportar la variable.
8. Elija Salidas.
9. En Variables (opcional), elija Añadir variable y, a continuación, haga lo siguiente:

Especifique el nombre de la variable que desee que se exporte la acción. Esta variable ya debe estar definida en la Steps sección Inputs o de la misma acción.

10. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
11. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

YAML

Para exportar una variable (editor YAML)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la acción desde la que desee exportar la variable, añada un código similar al siguiente:

```
action-name:  
  Outputs:  
    Variables:  
      - Name: variable-name
```

Para obtener más ejemplos, consulte [Ejemplos](#).

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Hacer referencia a una variable en la acción que la define

Siga las instrucciones siguientes para hacer referencia a una variable en la acción que la define.

Note

Para hacer referencia a una variable generada por una GitHub acción, consulte [Hacer referencia a un parámetro GitHub de salida](#).

Requisito previo

Asegúrese de haber definido la variable a la que desea hacer referencia. Para obtener más información, consulte [Definir una variable](#).

Visual

No disponible. Selecciona YAML para ver las instrucciones de YAML.

YAML

Para hacer referencia a una variable en la acción que la define

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la CodeCatalyst acción que define la variable a la que quieres hacer referencia, añade la variable con la siguiente sintaxis de bash:

```
$variable-name
```

Por ejemplo:

```
MyAction:
  Configuration:
    Steps:
      - Run: $variable-name
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener más información, consulte la información de referencia de su acción en [Referencia de definición de flujo de trabajo](#)

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Hacer referencia a una salida variable mediante otra acción

Utilice las siguientes instrucciones para hacer referencia a las variables generadas por otras acciones.

Note

Para hacer referencia a la salida de una variable de una GitHub acción, consulte [Hacer referencia a un parámetro GitHub de salida](#).

Requisito previo

Asegúrese de haber exportado la variable a la que desea hacer referencia. Para obtener más información, consulte [Exportar una variable para que otras acciones puedan utilizarla](#).

Visual

No disponible. Selecciona YAML para ver las instrucciones de YAML.

YAML

Para hacer referencia a la salida de una variable mediante otra acción (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la CodeCatalyst acción, añada una referencia a la variable mediante la siguiente sintaxis:

```
${action-group-name.action-name.variable-name}
```

Sustituir:

- *action-group-name* por el nombre del grupo de acciones que contiene la acción que genera la variable.

Note

Puede omitir *action-group-name* si no hay ningún grupo de acciones o si la variable es producida por una acción del mismo grupo de acciones.

- *action-name* con el nombre de la acción que genera la variable.
- *nombre-variable con el nombre* de la variable.

Por ejemplo:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: ${MyFirstAction.TIMESTAMP}
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener más información, consulte la sección [Referencia de definición de flujo de trabajo](#) para su acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Hacer referencia a un secreto

Para obtener instrucciones sobre cómo hacer referencia a un secreto en el archivo de definición del flujo de trabajo, consulte [Uso de un secreto](#)

Para ver un ejemplo, consulte [Ejemplo: hacer referencia a un secreto](#).

Ejemplos

Los siguientes ejemplos muestran cómo definir y hacer referencia a las variables en el archivo de definición del flujo de trabajo.

Ejemplos

- [Ejemplo: definir una variable mediante la propiedad Inputs](#)
- [Ejemplo: definir una variable mediante la propiedad Steps](#)

- [Ejemplo: exportar una variable mediante la propiedad Outputs](#)
- [Ejemplo: hacer referencia a una variable definida en la misma acción](#)
- [Ejemplo: hacer referencia a una variable definida en otra acción](#)
- [Ejemplo: hacer referencia a un secreto](#)

Ejemplo: definir una variable mediante la propiedad Inputs

El siguiente ejemplo muestra cómo definir dos variables VAR1 y VAR2, en una Inputs sección.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: "My variable 1"
        - Name: VAR2
          Value: "My variable 2"
```

Ejemplo: definir una variable mediante la propiedad Steps

El siguiente ejemplo muestra cómo definir una DATE variable en la Steps sección de forma explícita.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: DATE=$(date +%m-%d-%y)
```

Ejemplo: exportar una variable mediante la propiedad Outputs

El siguiente ejemplo muestra cómo definir dos variables REPOSITORY-URI y TIMESTAMP exportarlas mediante la Outputs sección.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
```

```

Variables:
  - Name: REPOSITORY-URI
    Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
Configuration:
  Steps:
    - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
  Outputs:
    Variables:
      - REPOSITORY-URI
      - TIMESTAMP

```

Ejemplo: hacer referencia a una variable definida en la misma acción

En el siguiente ejemplo, se muestra cómo especificar una VAR1 variable en la misma acción yMyBuildAction, a continuación, hacer referencia a ella en la misma acción mediante\$VAR1.

```

Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: my-value
    Configuration:
      Steps:
        - Run: $VAR1

```

Ejemplo: hacer referencia a una variable definida en otra acción

El siguiente ejemplo muestra cómo especificar una TIMESTAMP variable enBuildActionA, exportarla mediante la Outputs propiedad y, a continuación, hacer referencia a ella al BuildActionB \${BuildActionA.TIMESTAMP} utilizarla.

```

Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
    Outputs:
      Variables:

```

```
- TIMESTAMP
BuildActionB:
  Identifier: aws/build@v1
  Configuration:
    Steps:
      - Run: docker build -t my-ecr-repo/image-repo:latest .
      - Run: docker tag my-ecr-repo/image-repo:${BuildActionA.TIMESTAMP}

# Specifying just '$TIMESTAMP' here will not work
# because TIMESTAMP is not a variable
# in the BuildActionB action.
```

Ejemplo: hacer referencia a un secreto

En el siguiente ejemplo, se muestra cómo hacer referencia a un `my-password` secreto. `my-password` es la clave del secreto. La clave de este secreto y el valor de la contraseña correspondiente deben especificarse en la página de secretos de la CodeCatalyst consola antes de utilizarlos en el archivo de definición del flujo de trabajo. Para obtener más información, consulte [Trabajando con secretos](#).

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: curl -u LiJuan:${Secrets.my-password} https://example.com
```

Uso de variables predefinidas

Las variables predefinidas son pares clave-valor que un flujo de trabajo emite automáticamente y que están disponibles para su uso en las acciones del flujo de trabajo.

Puede utilizar variables predefinidas en cualquier acción del flujo de trabajo.

Temas

- [Lista de variables predefinidas](#)
- [Hacer referencia a una variable predefinida](#)
- [Determinar qué variables predefinidas emite tu flujo de trabajo](#)
- [Ejemplos](#)

Lista de variables predefinidas

Consulte las siguientes secciones para ver las variables predefinidas generadas automáticamente por CodeCatalyst.

Note

Esta lista solo incluye las variables predefinidas emitidas por los CodeCatalyst flujos de trabajo y [CodeCatalyst las acciones](#). Si utilizas otros tipos de acciones, como las acciones de GitHub Actions o CodeCatalyst Labs, consulta en su lugar [Determinar qué variables predefinidas emite tu flujo de trabajo](#).

Enumeración

- [variables BranchName «» y CommitId «»](#)
- [Cree y pruebe variables de acción](#)
- [Variables de acción «Amazon S3 publish»](#)
- [Variables de acción «AWS CDKbootstrap»](#)
- [«AWS CDKdesplegar» variables de acción](#)
- [«AWS Lambdainvocar» variables de acción](#)
- [Variables de acción «Desplegar AWS CloudFormation pila»](#)
- [Variables de acción «Implementar en Amazon ECS»](#)
- [Variables de acción «Desplegar en un clúster de Kubernetes»](#)
- [Variables de acción «Renderizar la definición de tareas de Amazon ECS»](#)
- [Variables de acción «GitHub Acciones»](#)

variables BranchName «» y CommitId «»

CodeCatalyst produce variables «BranchName» y «CommitId» cuando se ejecuta el flujo de trabajo. Consulte la siguiente tabla para obtener información sobre estas variables.

Clave	Valor
CommitId	El ID de confirmación que representa el estado del repositorio en el momento en que se inició la ejecución del flujo de trabajo.

Clave	Valor
	<p>Ejemplo: <code>example3819261db00a3ab59468c8b</code></p> <p>Véase también: Ejemplo: hacer referencia a la variable predefinida CommitId «»</p>
BranchName	<p>El nombre de la rama desde la que se inició la ejecución del flujo de trabajo.</p> <p>Ejemplos: <code>main</code>, <code>feature/branch</code> y <code>test-LiJuan</code></p> <p>Véase también: Ejemplo: hacer referencia a la variable predefinida BranchName «»</p>

Cree y pruebe variables de acción

Las acciones de compilación y prueba no generan variables automáticamente.

Variables de acción «Amazon S3 publish»

La acción de publicación de Amazon S3 no genera variables automáticamente.

Variables de acción «AWS CDKbootstrap»

La acción AWS CDKbootstrap produce las siguientes variables predefinidas.

Clave	Valor
plataforma de despliegue	<p>El nombre de la plataforma de despliegue.</p> <p>Codificado para. <code>AWS:CloudFormation</code></p>
region	<p>El código de región en el Región de AWS que se implementó la pila de AWS CDK bootstrap durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>us-west-2</code></p>

Clave	Valor
stack-id	<p>El nombre de recurso de Amazon (ARN) de la pila de AWS CDK bootstrap implementada.</p> <p>Ejemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/decatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code></p>
OMITIR EL DESPLIEGUE	<p>Un valor de <code>true</code> indica que se omitió el despliegue de la pila de AWS CDK bootstrap durante la ejecución del flujo de trabajo. Se omitirá el despliegue de una pila si no se ha producido ningún cambio en la pila desde la última implementación.</p> <p>Esta variable solo se genera si su valor es <code>true</code>.</p> <p>Codificado para. <code>true</code></p>

«AWS CDKdesplegar» variables de acción

La acción de AWS CDKdespliegue produce las siguientes variables.

Clave	Valor
stack-id	<p>El nombre de recurso de Amazon (ARN) de la pila de AWS CDK aplicaciones en la que se implementó durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/decatalyst-cdk-app-stack/6a</code></p>

Clave	Valor
	<code>ad4380-100a-11ec-a10a-03b8a84d40df</code>
plataforma de despliegue	El nombre de la plataforma de despliegue. Codificado para. <code>AWS:CloudFormation</code>
region	El código de región en el Región de AWS que se implementó durante la ejecución del flujo de trabajo. Ejemplo: <code>us-west-2</code>
OMITIR EL DESPLIEGUE	Un valor de <code>true</code> indica que se omitió el despliegue de la pila de AWS CDK aplicaciones durante la ejecución del flujo de trabajo. Se omitirá el despliegue de una pila si no se ha producido ningún cambio en la pila desde la última implementación. Esta variable solo se genera si su valor es <code>true</code> . Codificado para. <code>true</code>

Clave	Valor
AWS CloudFormationvariables	Además de generar las variables enumeradas anteriormente, la acción de AWS CDKdespliegue también expone las variables de CloudFormationsalida como variables de flujo de trabajo para su uso en acciones de flujo de trabajo posteriores. De forma predeterminada, la acción solo expone las cuatro primeras CloudFormation variables (o menos) que encuentre. Para determinar cuáles están expuestas, ejecute la acción de AWS CDKdespliegue una vez y, a continuación, consulte la pestaña Variables de la página de detalles de la ejecución. Si las variables que aparecen en la pestaña Variables no son las que deseas, puedes configurar otras variables mediante la propiedad CfnOutput Variables YAML. Para obtener más información, consulte la descripción de la CfnOutputVariables propiedad en. referencia de acción «AWS CDK desplegar»

«AWS Lambdainvocar» variables de acción

De forma predeterminada, la acción de AWS Lambdainvocación produce una variable por clave de nivel superior en la carga útil de respuesta de Lambda.

Por ejemplo, si la carga útil de la respuesta tiene este aspecto:

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

... entonces la acción generaría las siguientes variables.

Clave	Valor
nombre	Saanvi
location	Seattle
department	{"empresa": «Amazon», «equipo»: "AWS«}

Note

Puedes cambiar las variables que se generan mediante la propiedad `ResponseFilters` YAML. Para obtener más información, consulte [ResponseFilters](#) en la [referencia de acción «AWS Lambda invocar»](#).

Variables de acción «Desplegar AWS CloudFormation pila»

La acción Desplegar AWS CloudFormation pila produce las siguientes variables.

Clave	Valor
plataforma de despliegue	El nombre de la plataforma de despliegue. Codificado para. <code>AWS:CloudFormation</code>
region	El código de región en el Región de AWS que se implementó durante la ejecución del flujo de trabajo. Ejemplo: <code>us-west-2</code>
stack-id	El nombre de recurso de Amazon (ARN) de la pila implementada. Ejemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/co</code>

Clave	Valor
	<code>decatalyst-cfn-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>

Variables de acción «Implementar en Amazon ECS»

La acción Deploy to Amazon ECS produce las siguientes variables.

Clave	Valor
clúster	<p>El nombre del clúster de Amazon ECS en el que se implementó durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>codecatalyst-ecs-cluster</code></p>
plataforma de despliegue	<p>El nombre de la plataforma de despliegue.</p> <p>Codificado para. <code>AWS:ECS</code></p>
service	<p>El nombre del servicio Amazon ECS en el que se implementó durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>codecatalyst-ecs-service</code></p>
task-definition-arn	<p>El nombre del recurso de Amazon (ARN) de la definición de tarea que se registró durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</code></p> <p>El <code>:8</code> ejemplo anterior indica la revisión que se registró.</p>
url de despliegue	<p>Un enlace a la pestaña Eventos de la consola Amazon ECS, donde puede ver los detalles de</p>

Clave	Valor
	<p>la implementación de Amazon ECS asociada a la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>https://console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/codecatalyst-ecs-cluster/services/codecatalyst-ecs-service/events</code></p>
region	<p>El código de región en el Región de AWS que se implementó durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>us-west-2</code></p>

Variables de acción «Desplegar en un clúster de Kubernetes»

La acción Implementar en un clúster de Kubernetes produce las siguientes variables.

Clave	Valor
clúster	<p>El nombre de recurso (ARN) de Amazon.com del clúster de Amazon EKS en el que se implementó durante la ejecución del flujo de trabajo.</p> <p>Ejemplo: <code>arn:aws:eks:us-west-2:111122223333:cluster/codecatalyst-eks-cluster</code></p>
plataforma de despliegue	<p>El nombre de la plataforma de despliegue.</p> <p>Codificado para. <code>AWS:EKS</code></p>
metadatos	<p>Reservado. Metadatos con formato JSON relacionados con el clúster implementado durante la ejecución del flujo de trabajo.</p>

Clave	Valor
espacio de nombres	El espacio de nombres de Kubernetes en el que se implementó el clúster. Ejemplo: default
recursos	Reservado. Metadatos con formato JSON relacionados con los recursos desplegados durante la ejecución del flujo de trabajo.
servidor	El nombre del punto final del servidor API que puede usar para comunicarse con su clúster mediante herramientas de administración como. kubectl Para obtener más información sobre el punto de enlace del servicio de API, consulte el control de acceso al punto final del clúster de Amazon EKS en la Guía del usuario de Amazon EKS. Ejemplo: https:// <i>random-string</i> .gr7.us-west-2.eks.amazonaws.com

Variables de acción «Renderizar la definición de tareas de Amazon ECS»

La acción de definición de tareas Render Amazon ECS produce las siguientes variables.

Clave	Valor
definición de tareas	Nombre dado al archivo de definición de tareas que se actualizó mediante la acción de definición de tareas Render Amazon ECS. El nombre sigue este formato: task-definition- <i>random-string</i> .json.

Clave	Valor
	Ejemplo: task-definition--259-0a2r7gx1TF5Xr.json

Variables de acción «GitHub Acciones»

La acción «GitHub Acciones» no genera variables automáticamente.

Hacer referencia a una variable predefinida

Utilice las siguientes instrucciones para hacer referencia a una variable predefinida.

Requisito previo

Determine el nombre de la variable predefinida a la que desea hacer referencia, por ejemplo `CommitId`. Para obtener más información, consulte [Determinar qué variables predefinidas emite tu flujo de trabajo](#).

Visual

No disponible. Selecciona YAML para ver las instrucciones de YAML.

YAML


Para hacer referencia a una variable predefinida (editor YAML)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En una CodeCatalyst acción, añada la referencia de variable predefinida mediante la siguiente sintaxis:

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```


Sustituir:

- *action-group-name* por el nombre del grupo de acciones.

 Note

Puede omitir *action-group-name* si no hay ningún grupo de acciones o si la variable es producida por una acción del mismo grupo de acciones.

- *action-name-or- WorkflowSource con:*

El nombre de la acción que genera la variable.

o bien

WorkflowSource, si la variable es la CommitId variable BranchName o.

- nombre de la *variable con el nombre* de la variable.

Por ejemplo:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${MyFirstECSAction.cluster}
```

Otro ejemplo.

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${WorkflowSource.CommitId}
```

Para obtener más ejemplos, consulte [Ejemplos](#). Para obtener más información, consulte la sección [Referencia de definición de flujo de trabajo](#) para su acción.

8. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.

Determinar qué variables predefinidas emite tu flujo de trabajo

Puede determinar qué variables predefinidas emite su flujo de trabajo de dos maneras:

- Ejecute el flujo de trabajo una vez. Una vez finalizada la ejecución, las variables emitidas por el flujo de trabajo se muestran en la pestaña Variables de la página de detalles de la ejecución. Para obtener más información, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).
- Consulte el [Lista de variables predefinidas](#). Esta referencia muestra el nombre de la variable (clave) y el valor de cada variable predefinida.

Note

El tamaño total máximo de las variables de un flujo de trabajo aparece en [Cuotas para flujos de trabajo en CodeCatalyst](#). Si el tamaño total supera el máximo, es posible que se produzca un error en la acción que se produzca una vez alcanzado el máximo.

Ejemplos

Los siguientes ejemplos muestran cómo hacer referencia a variables predefinidas en el archivo de definición del flujo de trabajo.

Ejemplos

- [Ejemplo: hacer referencia a la variable predefinida CommitId «»](#)
- [Ejemplo: hacer referencia a la variable predefinida BranchName «»](#)

Ejemplo: hacer referencia a la variable predefinida CommitId «»

El siguiente ejemplo muestra cómo hacer referencia a la variable `CommitId` predefinida en la `MyBuildAction` acción. La `CommitId` variable se genera automáticamente mediante CodeCatalyst.

Aunque en el ejemplo se muestra la variable que se utiliza en la acción de creación, se puede utilizar `CommitId` en cualquier acción.

```
MyBuildAction:
  Identifier: aws/build@v1
  Inputs:
```

```

Sources:
  - WorkflowSource
Configuration:
Steps:
#Build Docker image and tag it with a commit ID
  - Run: docker build -t image-repo/my-docker-image:latest .
  - Run: docker tag image-repo/my-docker-image:${WorkflowSource.CommitId}

```

Ejemplo: hacer referencia a la variable predefinida BranchName «»

El siguiente ejemplo muestra cómo hacer referencia a la variable BranchName predefinida en la CDKDeploy acción. La BranchName variable se genera automáticamente mediante CodeCatalyst.

Aunque en el ejemplo se muestra la variable que se utiliza en la acción de AWS CDKdespliegue, se puede utilizar BranchName en cualquier acción.

```

CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    StackName: app-stack-${WorkflowSource.BranchName}

```

Referencia de definición de flujo de trabajo

La siguiente es la documentación de referencia para el archivo de definición del flujo de trabajo.

Un archivo de definición de flujo de trabajo es un archivo YAML que describe su flujo de trabajo. El archivo se almacena en una `~/codecatalyst/workflows/` carpeta en la raíz del [repositorio de origen](#). El archivo puede tener la extensión `.yml` o `.yaml`.

Para crear y editar el archivo de definición del flujo de trabajo, puedes usar un editor como vim, o puedes usar el editor visual o el editor YAML de la CodeCatalyst consola. Para obtener más información, consulte [Uso de los CodeCatalyst editores visuales y YAML de la consola](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los correspondientes elementos de interfaz de usuario en el editor visual. Para buscar un

elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

Temas

- [Ejemplo de un archivo de definición de flujo de trabajo](#)
- [Pautas y convenciones de sintaxis](#)
- [Propiedades de primer nivel](#)
- [Referencia de acciones de compilación y prueba](#)
- [Referencia de acción «Amazon S3 publish»](#)
- [referencia de acción «AWS CDKbootstrap»](#)
- [referencia de acción «AWS CDK desplegar»](#)
- [referencia de acción «AWS Lambda invocar»](#)
- [Referencia de acción «Implementar AWS CloudFormation pila»](#)
- [Referencia de acción «Implementar en Amazon ECS»](#)
- [Referencia de acción «Implementar en un clúster de Kubernetes»](#)
- [Referencia de acción «GitHub Acciones»](#)
- [Referencia de acción «Renderizar la definición de tareas de Amazon ECS»](#)

Ejemplo de un archivo de definición de flujo de trabajo

El siguiente es un ejemplo de un archivo de definición de flujo de trabajo sencillo. Incluye algunas propiedades de nivel superior, una `Triggers` sección y una `Actions` sección con dos acciones: `Build` y `Test`. Para obtener más información, consulte [Acerca del archivo de definición del flujo de trabajo](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  Build:
```

```
Identifier: aws/build@v1
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Steps:
    - Run: docker build -t MyApp:latest .
Test:
Identifier: aws/managed-test@v1
DependsOn:
  - Build
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Steps:
    - Run: npm install
    - Run: npm run test
```

Pautas y convenciones de sintaxis

En esta sección se describen las reglas de sintaxis del archivo de definición del flujo de trabajo, así como las convenciones de nomenclatura utilizadas en esta documentación de referencia.

Directrices de sintaxis de YAML

El archivo de definición del flujo de trabajo está escrito en YAML y sigue la [especificación YAML 1.1](#), por lo que todo lo que esté permitido en esa especificación también lo estará en el flujo de trabajo YAML. Si eres nuevo en YAML, aquí tienes algunas pautas rápidas para asegurarte de que estás proporcionando un código YAML válido.

- **Distingue entre mayúsculas y minúsculas:** el archivo de definición del flujo de trabajo distingue entre mayúsculas y minúsculas, así que asegúrate de usar las mayúsculas y minúsculas que se muestran en esta documentación.
- **Caracteres especiales:** se recomienda utilizar comillas o comillas dobles en los valores de las propiedades que incluyan alguno de los siguientes caracteres especiales: { } [] , , * , &# , ? , | , - , < , > , = , ! % , @ y : ` ,

Si no incluye las comillas, es posible que los caracteres especiales enumerados anteriormente se interpreten de forma inesperada.

- **Nombres de propiedades:** los nombres de las propiedades (a diferencia de los valores de las propiedades) están limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede utilizar comillas ni comillas dobles para activar los caracteres especiales y los espacios en los nombres de las propiedades.

No está permitido:

```
'My#Build@action'
```

```
My#Build@action
```

```
My Build Action
```

Permitido:

```
My-Build-Action_1
```

- **Códigos de escape:** si el valor de su propiedad incluye códigos de escape (por ejemplo, `\n` o `\t`), siga estas pautas:
 - Usa comillas simples para devolver el código de escape en forma de cadena. Por ejemplo `'my string \n my string'`, devuelve la cadena `my string \n my string`.
 - Use comillas dobles para analizar el código de escape. Por ejemplo `"my string \n my new line"`, devuelve:

```
my string
my new line
```

- **Comentarios:** los comentarios del prefacio incluyen. `#`

Ejemplo:

```
Name: MyWorkflow
# This is a comment.
SchemaVersion: 1.0
```

- **Triple dash (---):** no lo utilices `---` en tu código YAML. CodeCatalyst ignora todo lo que está después de. `---`

Convenciones de nomenclatura

En esta guía, utilizamos los términos propiedad y sección para referirnos a los elementos principales de un archivo de definición de flujo de trabajo.

- Una propiedad es cualquier elemento que incluye dos puntos (:). Por ejemplo, en el siguiente fragmento de código, todas las propiedades siguientes son: `Name`, `SchemaVersion`, `RunMode`, `TriggersType`, y `Branches`
- Una sección es cualquier propiedad que tiene subpropiedades. En el siguiente fragmento de código, hay una sección. `Triggers`

Note

En esta guía, las «secciones» a veces se denominan «propiedades» y viceversa, según el contexto.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

Propiedades de primer nivel

La siguiente es la documentación de referencia para las propiedades de nivel superior del archivo de definición del flujo de trabajo.

```
# Name
Name: workflow-name

# Schema version
SchemaVersion: 1.0

# Run mode
RunMode: QUEUED|SUPERSEDED|PARALLEL
```

```
# Compute
Compute:
...

# Triggers
Triggers:
...

# Actions
Actions:
...
```

Name

(Obligatorio)

Nombre del flujo de trabajo. El nombre del flujo de trabajo se muestra en la lista de flujos de trabajo y se menciona en las notificaciones y los registros. El nombre del flujo de trabajo y el nombre del archivo de definición del flujo de trabajo pueden coincidir, o puede asignarles un nombre diferente. Los nombres de los flujos de trabajo no tienen por qué ser únicos. Los nombres de los flujos de trabajo se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de los flujos de trabajo.

Interfaz de usuario correspondiente: editor visual/propiedades del flujo de trabajo/nombre del flujo de trabajo

SchemaVersion

(Obligatorio)

La versión esquemática de la definición del flujo de trabajo. El único valor válido actualmente es 1.0.

Interfaz de usuario correspondiente: ninguna

RunMode

(Opcional)

Cómo CodeCatalyst gestiona las ejecuciones múltiples. Puede utilizar uno de los siguientes valores:

- **QUEUED**— Se ponen en cola varias corridas y se ejecutan una tras otra. Puedes tener hasta 50 carreras en una cola.
- **SUPERSEDED**— Se ponen varias carreras en cola y se ejecutan una tras otra. Una cola solo puede tener una ejecución, por lo que si dos corridas terminan juntas en la misma cola, la última reemplaza (reemplaza) a la anterior y esta última se cancela.
- **PARALLEL**— Se producen varias corridas simultáneamente.

Si se omite esta propiedad, el valor por defecto es **QUEUED**.

Para obtener más información, consulte [Configuración de ejecuciones en cola, reemplazadas y paralelas](#).

Interfaz de usuario correspondiente: editor visual/Propiedades del flujo de trabajo/Avanzada/Modo de ejecución

Compute

(Opcional)

El motor informático que se utiliza para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Para obtener más información sobre el procesamiento, consulte [Trabajar con imágenes de Docker de entornos de procesamiento y tiempo de ejecución](#).

Interfaz de usuario correspondiente: ninguna

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
  SharedInstance: true | false
```

Type

(Compute/Type)

(Obligatorio si Compute está configurado)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: editor visual/Propiedades del flujo de trabajo/Avanzada/Tipo de cómputo

Fleet

(Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. `Linux.x86-64.Large`

Para obtener más información sobre las flotas informáticas, consulte. [Acerca de las flotas informáticas](#)

Interfaz de usuario correspondiente: editor visual/Propiedades del flujo de trabajo/Avanzada/Flota de cómputo

SharedInstance

(Compute/SharedInstance)

(Opcional)

Especifique la capacidad de compartir el procesamiento para sus acciones. Con el uso compartido de recursos informáticos, las acciones de un flujo de trabajo se ejecutan en la misma instancia (imagen del entorno de ejecución). Puedes usar uno de los siguientes valores:

- **TRUE** significa que la imagen del entorno de ejecución se comparte entre las acciones del flujo de trabajo.
- **FALSE** significa que se inicia una imagen del entorno de ejecución independiente y se utiliza para cada acción de un flujo de trabajo, por lo que no se pueden compartir recursos como artefactos y variables sin una configuración adicional.

Para obtener más información sobre el uso compartido de la computación, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Triggers

(Opcional)

Secuencia de uno o más activadores de este flujo de trabajo. Si no se especifica ningún desencadenante, debe iniciar el flujo de trabajo manualmente.

Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores

```
Name: MyWorkflow
SchemaVersion: 1.0
...
```

Triggers:

- **Type:** PUSH

Branches:

- *branch-name*

FilesChanged:

- folder1/file
- folder2/

- **Type:** PULLREQUEST

Events:

- *OPEN*
- *CLOSED*
- *REVISION*

Branches:

- *branch-name*

FilesChanged:

- file1.txt

- **Type:** SCHEDULE

Run the workflow at 10:15 am (UTC+0) every Saturday

Expression: "15 10 ? * 7 *"

Branches:

- *branch-name*

Type

(Triggers/Type)

(Obligatorio si está configurado) Triggers

Especifique el tipo de disparador. Puede utilizar uno de los siguientes valores:

- Push (editor visual) o PUSH (editor YAML)

Un disparador push inicia la ejecución de un flujo de trabajo cuando se envía un cambio a tu repositorio de origen. La ejecución del flujo de trabajo utilizará los archivos de la rama a la que vayas a enviar (es decir, la rama de destino).

- Pull Request (editor visual) o PULLREQUEST (editor YAML)

Un activador de solicitudes de extracción inicia la ejecución de un flujo de trabajo cuando se abre, actualiza o cierra una solicitud de extracción en tu repositorio de origen. La ejecución del flujo de trabajo utilizará los archivos de la rama de la que estás extrayendo (es decir, la rama de origen).

- Programación (editor visual) o SCHEDULE (editor YAML)

Un activador de programación inicia las ejecuciones del flujo de trabajo según una programación definida por una expresión cron que usted especifique. Se iniciará una ejecución de flujo de trabajo independiente para cada rama del repositorio de origen utilizando los archivos de la rama. (Para limitar las ramas en las que se activa el activador, usa el campo Ramas (editor visual) o la Branches propiedad (editor YAML).)

Al configurar un activador programado, siga estas pautas:

- Utilice solo un activador de programación por flujo de trabajo.
- Si ha definido varios flujos de trabajo en su CodeCatalyst espacio, le recomendamos que no programe más de 10 de ellos para que se inicien simultáneamente.
- Asegúrese de configurar la expresión cron del disparador con el tiempo adecuado entre ejecuciones. Para obtener más información, consulte [Expression](#).

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores/tipo de disparador

Events

(Triggers/Events)

(Obligatorio si el disparador está configurado en) Type PULLREQUEST

Especifica el tipo de eventos de solicitud de cambios que iniciarán la ejecución de un flujo de trabajo. Los valores válidos son los siguientes:

- Se crea una solicitud de extracción (editor visual) o OPEN (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se crea una solicitud de extracción.

- La solicitud de extracción está cerrada (editor visual) o CLOSED (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se cierra una solicitud de extracción. El comportamiento del CLOSED evento es complicado y se entiende mejor con un ejemplo. Para obtener más información, consulte [Ejemplo: un gatillo al que se tira, se ramifica y se activa el evento «CERRADO»](#).

- Se ha realizado una nueva revisión en pull request (editor visual) o REVISION (editor YAML)

La ejecución del flujo de trabajo se inicia cuando se crea una revisión de una solicitud de extracción. La primera revisión se crea cuando se crea la solicitud de extracción. Después de eso, se crea una nueva revisión cada vez que alguien envía una nueva confirmación a la rama de origen especificada en la solicitud de extracción. Si incluyes el REVISION evento en el activador de la solicitud de cambios, puedes omitirlo, ya que REVISION es un superconjunto de. OPEN OPEN

Puedes especificar varios eventos en el mismo activador de la solicitud de atracción.

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores/eventos para la solicitud de extracción

Branches

(Triggers/Branches)

(Opcional)

Especifica las ramas del repositorio de origen que monitorea el activador para saber cuándo iniciar la ejecución de un flujo de trabajo. Puedes usar patrones de expresiones regulares para definir los nombres de tus ramas. Por ejemplo, úsalo `main.*` para hacer coincidir todas las ramas que comiencen por. `main`

Las ramas que se deben especificar son diferentes en función del tipo de disparador:

- En el caso de un disparador automático, especifique las ramas a las que va a enviar el mensaje, es decir, las sucursales de destino. Se iniciará una ejecución de flujo de trabajo por cada sucursal coincidente, utilizando los archivos de la sucursal coincidente.

Ejemplos: `main.*`, `mainline`

- Para activar una solicitud de cambios, especifica las sucursales a las que vas a enviar mensajes, es decir, las sucursales de destino. Se iniciará una ejecución de flujo de trabajo por cada rama coincidente, utilizando el archivo de definición del flujo de trabajo y los archivos fuente de la rama de origen (no de la rama coincidente).

Ejemplos: `main.*`, `mainline`, `v1\-.*` (coincide con las ramas que comienzan con `v1-`)

- Para activar una programación, especifique las ramas que contienen los archivos que desea que utilice la ejecución programada. Se iniciará una ejecución de flujo de trabajo por cada rama

coincidente, utilizando el archivo de definición del flujo de trabajo y los archivos fuente de la rama coincidente.

Ejemplos: `main.*`, `version*-1\.*`

Note

Si no especificas las ramas, el activador supervisa todas las ramas del repositorio de origen e iniciará una ejecución de flujo de trabajo con el archivo de definición del flujo de trabajo y los archivos fuente de:

- La rama hacia la que estás presionando (para los activadores de pulsación). Para obtener más información, consulte [Ejemplo: un sencillo disparador por pulsación de código](#).
- La sucursal desde la que realizas la extracción (para activar las solicitudes de extracción). Para obtener más información, consulte [Ejemplo: un sencillo activador de una solicitud de extracción](#).
- Todas las sucursales (para programar los activadores). Se iniciará una ejecución de flujo de trabajo por cada rama del repositorio de origen. Para obtener más información, consulte [Ejemplo: un sencillo disparador programado](#).

Para obtener más información sobre las ramas y los activadores, consulte [Consideraciones sobre los factores desencadenantes al ramificar](#).

Para obtener más ejemplos, consulte [Ejemplos de desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores/ramas

FilesChanged

(Triggers/FilesChanged)

(Opcional si el disparador está configurado en, o. Type PUSH PULLREQUEST No se admite si el disparador Type está configurado en SCHEDULE.

Especifique los archivos o carpetas del repositorio de origen que supervisa el desencadenador para saber cuándo iniciar la ejecución de un flujo de trabajo. Puede utilizar expresiones regulares para hacer coincidir los nombres o las rutas de los archivos.

Para ver ejemplos, consulte [Ejemplos de desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores/archivos modificados

Expression

(Triggers/Expression)

(Obligatorio si el disparador está configurado en) Type SCHEDULE

Especifique la expresión cron que describe cuándo desea que se ejecute el flujo de trabajo programado.

Las expresiones cron utilizadas CodeCatalyst utilizan la siguiente sintaxis de seis campos, en la que cada campo está separado por un espacio:

minutos , horas days-of-month, mes, año. days-of-week

Ejemplos de expresiones cron

Minutos	Horas	Días del mes	Mes	Días de la semana	Año	Significado
0	0	?	*	MON-FRI	*	Ejecuta un flujo de trabajo a medianoche (UTC+0) de lunes a viernes.
0	2	*	*	?	*	Ejecuta un flujo de trabajo a las 2:00 a. m. (UTC +0) todos los días.
15	22	*	*	?	*	Ejecuta un flujo de

Minutos	Horas	Días del mes	Mes	Días de la semana	Año	Significado
						trabajo a las 22:15 (UTC+0) todos los días.
0/30	22-2	?	*	SAT-SUN	*	Ejecuta un flujo de trabajo cada 30 minutos de sábado a domingo, entre las 22:00 del día de inicio y las 2:00 del día siguiente (UTC+0).

Minutos	Horas	Días del mes	Mes	Días de la semana	Año	Significado
45	13	L	*	?	2023-2027	Ejecuta un flujo de trabajo a las 13:45 (UTC+0) del último día del mes entre los años 2023 y 2027, ambos inclusive.

Al especificar las expresiones cron CodeCatalyst, asegúrate de seguir estas pautas:

- Especifique una sola expresión cron por SCHEDULE activador.
- Escribe la expresión cron entre comillas dobles (") en el editor YAML.
- Especifique la hora en hora universal coordinada (UTC). No se admiten otras zonas horarias.
- Configure al menos 30 minutos entre ejecuciones. No se admite una cadencia más rápida.
- Especifique el *days-of-week* campo *days-of-month*, pero no ambos. Si especifica un valor o un asterisco (*) en uno de los campos, debe utilizar un signo de interrogación (?) en el otro. El asterisco significa «todos» y el signo de interrogación significa «cualquiera».

Para obtener más ejemplos de expresiones cron e información sobre caracteres comodín?, como, y *L, consulte la [referencia sobre expresiones cron en](#) la Guía del usuario de Amazon EventBridge . Las expresiones cron CodeCatalyst funcionan exactamente de EventBridge la misma manera.

Para ver ejemplos de activadores de horarios, consulte [Ejemplos de desencadenantes](#).

Interfaz de usuario correspondiente: editor visual/diagrama de flujo de trabajo/activadores/ programación

Acciones

Secuencia de una o más acciones para este flujo de trabajo. CodeCatalyst admite varios tipos de acciones, como las de creación y prueba, que ofrecen distintos tipos de funciones. Cada tipo de acción tiene:

- una `Identifier` propiedad que indica el identificador único y codificado de la acción. Por ejemplo, `aws/build@v1` identifica la acción de creación.
- una `Configuration` sección que contiene propiedades específicas de la acción.

Para obtener más información sobre cada tipo de acción, consulte su documentación de referencia, disponible en [Referencia de definición de flujo de trabajo](#).

La siguiente es la referencia de YAML para las acciones y los grupos de acciones del archivo de definición del flujo de trabajo.

Para obtener más información sobre las acciones, consulte [Trabajar con acciones](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Actions:
  action-name:
    Identifier: action-identifier
    Configuration:
      ...
  #Action groups
  action-group-name:
    Actions:
      ...
```

nombre de la acción

(nombre de la acción) Actions/

(Obligatorio)

Sustituya el *nombre de la acción* por el nombre que desee asignar a la acción. Los nombres de las acciones deben ser únicos en el flujo de trabajo y solo deben incluir caracteres alfanuméricos, guiones y guiones bajos. Para obtener más información sobre las reglas de sintaxis, consulte.

[Directrices de sintaxis de YAML](#)

Para obtener más información sobre las prácticas de nomenclatura de las acciones, incluidas las restricciones, consulte [la nombre de la acción](#).

Interfaz de usuario correspondiente: editor visual/nombre de la acción/pestaña de **configuración/Nombre de la acción** o nombre para mostrar de la acción

action-group-name

(Actions/*action-group-name*)

(Opcional)

Un grupo de acciones contiene una o más acciones. Agrupar las acciones en grupos de acciones le ayuda a mantener el flujo de trabajo organizado y también le permite configurar las dependencias entre los distintos grupos de acciones.

action-group-name Sustitúyalo por el nombre que desee asignar al grupo de acciones. Los nombres de los grupos de acciones deben ser únicos en el flujo de trabajo y solo deben incluir caracteres alfanuméricos, guiones y guiones bajos. Para obtener más información sobre las reglas de sintaxis, consulte [Directrices de sintaxis de YAML](#).

Para obtener más información sobre los grupos de acciones, consulte [Agrupación de acciones en grupos de acciones](#).

Interfaz de usuario correspondiente: ninguna

Referencia de acciones de compilación y prueba

La siguiente es la referencia de YAML de la definición de acciones para las acciones de compilación y prueba. Hay una referencia para dos acciones porque sus propiedades YAML son muy similares.

Elige una propiedad YAML en el siguiente código para ver su descripción.

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
```

```
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/build@v1 | aws/managed-test@v1
  DependsOn:
    - dependent-action-name-1
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Caching:
    FileCaching:
      key-name-1:
        Path: file1.txt
        RestoreKeys:
          - restore-key-1
  Inputs:
    Sources:
      - source-name-1
      - source-name-2
    Artifacts:
      - artifact-name
  Variables:
    - Name: variable-name-1
      Value: variable-value-1
    - Name: variable-name-2
      Value: variable-value-2
  Outputs:
    Artifacts:
      - Name: output-artifact-1
        Files:
          - build-output/artifact-1.jar
          - "build-output/build*"
      - Name: output-artifact-2
```

Files:

- build-output/artifact-2.1.jar
- build-output/artifact-2.2.jar

Variables:

- *variable-name-1*
- *variable-name-2*

AutoDiscoverReports:

Enabled: *true | false*

ReportNamePrefix: *AutoDiscovered*

IncludePaths:

- ***/**

ExcludePaths:

- *node_modules/cdk/junit.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisBug:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisSecurity:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisQuality:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisFinding:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

Reports:*report-name-1*:

Format: *format*

IncludePaths:

- **.xml*

ExcludePaths:

- *report2.xml*
- *report3.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisBug:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisSecurity:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisQuality:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisFinding:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*Configuration:Container:Registry: *registry*Image: *image*Steps:- Run: *"step 1"*- Run: *"step 2"*Packages:NpmConfiguration:PackageRegistries:- PackageRepository: *package-repository*Scopes:- *"@scope"*

nombre de la acción

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de la acción

Identifier

(nombre de *acción*/) Identifier

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Se usa `aws/build@v1` para acciones de construcción.

Se usa `aws/managed-test@v1` para acciones de prueba.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/nombre-acción/aws/build @v1 |aws/managed-test @v1 label

DependsOn

(nombre-acción/) DependsOn

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(nombre de *acción*/) Compute

(Opcional)

El motor informático que se utiliza para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(tipo de *nombre de acción*/Compute/)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/tipo de cálculo

Fleet

(nombre de la *acción* /*Compute*/ *Fleet*)

(Opcional)

Especifique la máquina o flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte [Propiedades de la flota aprovisionada](#)

Si `Fleet` se omite, el valor predeterminado es `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/flota de cómputo

Timeout

(nombre de *acción*/) Timeout

(Opcional)

Especifique la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que CodeCatalyst finalice la acción. El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Environment

(nombre de *acción*/) Environment

(Opcional)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/entorno

Name

(nombre de la *acción* /*Environment*/ *Name*)

(Opcional)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/nombre del entorno

Connections

(nombre de la *acción* /*Environment*/ *Connections*)

(Opcional)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo de Environment.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/

Name

(nombre de la *acción /Environment/Connections/ Name*)

(Opcional)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/

Role

(nombre de la *acción /Environment/Connections/ Role*)

(Opcional)

Especifique el nombre de la función de IAM que utiliza esta acción para acceder y operar en AWS servicios como Amazon S3 y Amazon ECR. Asegúrese de añadir esta función a la conexión de su cuenta. Para añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Es posible que pueda especificar el nombre del CodeCatalystWorkflowDevelopmentRole-*spaceName* rol aquí, siempre que tenga permisos suficientes. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Tenga en cuenta que el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.

Warning

Limite los permisos a los necesarios para las acciones de compilación y prueba. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

Interfaz de usuario correspondiente: pestaña de configuración/

Caching

(nombre-acción/) Caching

(Opcional)

Una sección en la que puede especificar una memoria caché para guardar los archivos en el disco y restaurarlos desde esa memoria caché en las siguientes ejecuciones del flujo de trabajo.

Para obtener más información sobre el almacenamiento en caché de archivos, consulte. [Trabajar con el almacenamiento en caché de archivos](#)

Interfaz de usuario correspondiente: pestaña de configuración/almacenamiento en caché de archivos: opcional

FileCaching

(nombre de la acción) /Caching/ FileCaching

(Opcional)

Sección que especifica la configuración de una secuencia de cachés.

Interfaz de usuario correspondiente: pestaña de configuración/Almacenamiento en caché de archivos (opcional)/Agregar caché

nombre-clave-1

(nombre-acción-nombre-clave-1) /Caching/FileCaching/

(Opcional)

Especifique el nombre de su propiedad de caché principal. Los nombres de las propiedades de la caché deben ser únicos en el flujo de trabajo. Cada acción puede tener hasta cinco entradasFileCaching.

Interfaz de usuario correspondiente: pestaña de configuración/almacenamiento en caché de archivos: opcional/Agregar caché/clave

Path

(/Caching/FileCaching/Pathnombre-acción-nombre-de-clave-1/)

(Opcional)

Especifique la ruta asociada a la memoria caché.

Interfaz de usuario correspondiente: pestaña de configuración/Almacenamiento en caché de archivos: opcional/Agregar caché/ruta

RestoreKeys

(/Caching/FileCaching/RestoreKeysnombre-acción-nombre-de-clave-1/)

(Opcional)

Especifique la clave de restauración para utilizarla como alternativa cuando no se pueda encontrar la propiedad de caché principal. Los nombres de las claves de restauración deben ser exclusivos de su flujo de trabajo. Cada caché puede contener hasta cinco entradasRestoreKeys.

Interfaz de usuario correspondiente: pestaña de configuración/Almacenamiento en caché de archivos: opcional/Agregar caché/Restaurar claves: opcional

Inputs

(Inputsnombre de la acción/)

(Opcional)

La Inputs sección define los datos que necesita una acción durante la ejecución de un flujo de trabajo.

Note

Se permite un máximo de cuatro entradas (una fuente y tres artefactos) por acción de construcción o acción de prueba. Las variables no se tienen en cuenta para este total.

Si necesita hacer referencia a archivos que se encuentran en entradas diferentes (por ejemplo, una fuente y un artefacto), la entrada de origen es la entrada principal y el artefacto es la entrada secundaria. Las referencias a los archivos en las entradas secundarias llevan un prefijo especial para distinguirlas de las principales. Para obtener más detalles, consulte [Ejemplo: hacer referencia a archivos en varios artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(nombre de *la acción /Inputs/ Sources*)

(Opcional)

Especifique las etiquetas que representan los repositorios de origen que necesitará la acción. Actualmente, la única etiqueta admitida es `WorkflowSource` la que representa el repositorio de origen en el que se almacena el archivo de definición de flujo de trabajo.

Si omite una fuente, debe especificar al menos un artefacto de entrada en ella. *action-name/Inputs/Artifacts*

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: ninguna

Artifacts - input

(nombre de *la acción /Inputs/ Artifacts*)

(Opcional)

Especifique los artefactos de acciones anteriores que desee proporcionar como entrada a esta acción. Estos artefactos ya deben estar definidos como artefactos de salida en acciones anteriores.

Si no especifica ningún artefacto de entrada, debe especificar al menos un repositorio de origen en *la acción/Inputs/Sources*.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Note

Si la lista desplegable Artefactos: opcional no está disponible (editor visual) o si se producen errores al validar el YAML (editor de YAML), es posible que la acción solo admita una entrada. En este caso, intenta eliminar la entrada de origen.

Interfaz de usuario correspondiente: pestaña de entradas/Artefactos (opcional)

Variables - input

(nombre de la *acción /Inputs/ Variables*)

(Opcional)

Especifique una secuencia de pares de nombre/valor que definan las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de entradas/variables: opcional

Outputs

(nombre de *acción/*) Outputs

(Opcional)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts - output

(nombre de *la acción /Outputs/ Artifacts*)

(Opcional)

Especifique el nombre de un artefacto generado por la acción. Los nombres de los artefactos deben ser únicos dentro de un flujo de trabajo y están limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) y guiones bajos (_). No se permiten espacios, guiones (-) ni otros caracteres especiales. No puede utilizar comillas para activar los espacios, guiones y otros caracteres especiales en los nombres de los artefactos de salida.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#)

Interfaz de usuario correspondiente: pestaña de salidas/artefactos

Name

(nombre de la *acción* /*Outputs/Artifacts/ Name*)

(Obligatorio si [Artifacts - output](#) está incluido)

Especifique el nombre de un artefacto generado por la acción. Los nombres de los artefactos deben ser únicos dentro de un flujo de trabajo y están limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) y guiones bajos (_). No se permiten espacios, guiones (-) ni otros caracteres especiales. No puede utilizar comillas para activar los espacios, guiones y otros caracteres especiales en los nombres de los artefactos de salida.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#)

Interfaz de usuario correspondiente: pestaña de salidas/Artifacts/Nueva salida/Construye el nombre del artefacto

Files

(*nombre de la acción*) /*Outputs/Artifacts/ Files*

(Obligatorio si [Artifacts - output](#) está incluido)

Especifique los archivos que se CodeCatalyst incluyen en el artefacto que genera la acción. Estos archivos los genera la acción del flujo de trabajo cuando se ejecuta y también están disponibles en el repositorio de origen. Las rutas de los archivos pueden residir en un repositorio de origen o en un artefacto de una acción anterior, y son relativas al repositorio de origen o a la raíz del artefacto. Puede utilizar patrones globales para especificar las rutas. Ejemplos:

- Para especificar un único archivo que esté en la raíz de la ubicación de compilación o de la ubicación del repositorio de origen, utilice `my-file.jar`.
- Para especificar un único archivo en un subdirectorio, use `directory/my-file.jar` o `directory/subdirectory/my-file.jar`.
- Para especificar todos los archivos, utilice `"/**/*"`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.
- Para especificar todos los archivos y directorios de un directorio denominado `directory`, utilice `"directory/**/*"`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.

- Para especificar todos los archivos de un directorio denominado `directory`, pero no ninguno de sus subdirectorios, utilice `"directory/*"`.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otro carácter especial, escriba la ruta entre comillas dobles (). "" Para obtener más información sobre los caracteres especiales, consulte [Pautas y convenciones de sintaxis](#)

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Note

Puede que tenga que añadir un prefijo a la ruta del archivo para indicar en qué artefacto o fuente puede encontrarlo. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña de salidas/Artefactos/Nueva salida/ Archivos producidos por la compilación

Variables - output

*(nombre de la acción) /Outputs/ **Variables***

(Opcional)

Especifique las variables que desea que exporte la acción para que estén disponibles para su uso en acciones posteriores.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de salidas/Variables/ Añadir variable

nombre de la variable-1

(nombre-de-acción-nombre-de-variable-1) /Outputs/Variables/

(Opcional)

Especifique el nombre de la variable que desee que exporte la acción. Esta variable ya debe estar definida en la Steps sección Inputs o de la misma acción.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de salidas/Variables/Añadir variable/ Nombre

AutoDiscoverReports

*(nombre de la acción) /Outputs/ **AutoDiscoverReports***

(Opcional)

Define la configuración de la función de detección automática.

Al habilitar la detección automática, CodeCatalyst busca todos los archivos Inputs transferidos a la acción, así como todos los archivos generados por la propia acción, buscando informes de pruebas, cobertura de código y análisis de composición de software (SCA). Para cada informe que se encuentre, lo CodeCatalyst transforma en un CodeCatalyst informe. Un CodeCatalyst informe es un informe que está totalmente integrado en el CodeCatalyst servicio y se puede ver y manipular a través de la CodeCatalyst consola.

Note

De forma predeterminada, la función de detección automática inspecciona todos los archivos. Puede limitar los archivos que se inspeccionan mediante las propiedades [IncludePaths](#) o [ExcludePaths](#).

Interfaz de usuario correspondiente: pestaña de salidas/informes/informes de detección automática

Enabled

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **Enabled***

(Opcional)

Activa o desactiva la función de detección automática.

Los valores válidos son `true` o `false`.

Si `Enabled` se omite, el valor predeterminado es `true`

Interfaz de usuario correspondiente: pestaña de salidas/informes/informes de detección automática

ReportNamePrefix

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **ReportNamePrefix***

(Obligatorio si [AutoDiscoverReports](#) está incluido y activado)

Especifique un prefijo que CodeCatalyst preceda a todos los informes que encuentre para asignar un nombre a los informes asociados. CodeCatalyst Por ejemplo, si especifica un prefijo de `AutoDiscovered` y CodeCatalyst descubre automáticamente dos informes de pruebas `TestSuiteOne.xml` y `TestSuiteTwo.xml`, los CodeCatalyst informes asociados se `AutoDiscoveredTestSuiteOne` denominarán y `AutoDiscoveredTestSuiteTwo`

Interfaz de usuario correspondiente: pestaña de resultados, informes y nombre del prefijo

IncludePaths

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **IncludePaths***

Or (Disyunción)

*(nombre-acción-nombre-de-informe-1//Outputs/Reports/) **IncludePaths***

(Obligatorio si [AutoDiscoverReports](#) está incluido y activado, o si está incluido) [Reports](#)

Especifique los archivos y las rutas de archivo que se CodeCatalyst incluyen al buscar informes sin procesar. Por ejemplo, si lo especificas `"/test/report/*"`, CodeCatalyst busca en el `/test/report/*` directorio toda la [imagen de compilación](#) utilizada por la acción. Cuando encuentra ese directorio, CodeCatalyst busca los informes en ese directorio.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otros caracteres especiales, escriba la ruta entre comillas dobles ("). "" Para obtener más información sobre los caracteres especiales, consulte. [Pautas y convenciones de sintaxis](#)

Si se omite esta propiedad, será la predeterminada "**/*/*", lo que significa que la búsqueda incluirá todos los archivos de todas las rutas.

 Note

En el caso de los informes configurados manualmente, `IncludePaths` debe ser un patrón global que coincida con un único archivo.

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Incluir/Excluir rutas/Incluir rutas
- Pestaña de salidas/Informes/Configurar informes manuales/nombre-informe-1 /Incluir/Excluir rutas/ **Incluir** rutas

ExcludePaths


(/Outputs/AutoDiscoverReports/ExcludePathsnombre-acción)

Or (Disyunción)

(nombre-acción-nombre-de-informe-1//Outputs/Reports/) ExcludePaths

(Opcional)

Especifique los archivos y las rutas de archivo que se CodeCatalyst excluyen al buscar informes sin procesar. Por ejemplo, si lo especifica `/test/my-reports/**/*/*`, no CodeCatalyst buscará archivos en el `/test/my-reports/` directorio. Para ignorar todos los archivos de un directorio, utilice el patrón `**/*/*` global.

 Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otros caracteres especiales, escriba la ruta entre comillas dobles ("). Para obtener más información sobre los caracteres especiales, consulte [Pautas y convenciones de sintaxis](#)

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Incluir/Excluir rutas/Excluir rutas
- Pestaña de salidas/Informes/Configurar informes manualmente/nombre-informe-1 /Include/Excluir rutas/ **Excluir** rutas

SuccessCriteria

(/Outputs/AutoDiscoverReports/SuccessCriteria nombre-acción)

Or (Disyunción)

*(nombre-acción-nombre-de-informe-1//Outputs/Reports/) **SuccessCriteria***

(Opcional)

Especifique los criterios de éxito de los informes de prueba, cobertura del código, análisis de composición de software (SCA) y análisis estático (SA).

Para obtener más información, consulte [Configurar los criterios de éxito de los informes](#).

Interfaz de usuario correspondiente: pestaña de salida/informes/criterios de éxito

PassRate

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/**PassRate***

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**PassRate***

(Opcional)

Especifique el porcentaje de pruebas en un informe de prueba que deben aprobarse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen números decimales. Por ejemplo: 50, 60.5. Los criterios de porcentaje de aprobados se aplican únicamente a los informes de las pruebas. Para obtener más información sobre los informes de las pruebas, consulte [Informes de pruebas](#).

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Tasa de aprobación

LineCoverage

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/
LineCoverage*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/
LineCoverage*

(Opcional)

Especifique el porcentaje de líneas de un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de línea se aplican únicamente a los informes de cobertura de código. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Cobertura de línea

BranchCoverage

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/
BranchCoverage*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/
BranchCoverage*

(Opcional)

Especifique el porcentaje de ramificaciones de un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de las sucursales se aplican únicamente a los informes de cobertura de códigos. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Cobertura de sucursales

Vulnerabilities

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/**Vulnerabilities***

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**Vulnerabilities***

(Opcional)

Especifique el número máximo y la gravedad de las vulnerabilidades permitidas en el informe de la SCA para que el informe asociado CodeCatalyst se marque como aprobado. Para especificar las vulnerabilidades, debe especificar:

- La gravedad mínima de las vulnerabilidades que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo eligeHIGH, se HIGH contabilizarán CRITICAL las vulnerabilidades.

- El número máximo de vulnerabilidades de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de vulnerabilidad se aplican únicamente a los informes de la SCA. Para obtener más información sobre los informes de la SCA, consulte. [Informes de análisis de composición de software](#)

Para especificar la gravedad mínima, utilice la Severity propiedad. Para especificar el número máximo de vulnerabilidades, utilice la Number propiedad.

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Vulnerabilidades

StaticAnalysisBug

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/**StaticAnalysisBug***

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**StaticAnalysisBug***

(Opcional)

Especifique el número máximo y la gravedad de los errores permitidos en el informe de SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar los errores, debe especificar:

- La gravedad mínima de los errores que quieres incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo eligesHIGH, se HIGH contabilizarán CRITICAL los errores.

- El número máximo de errores de la gravedad especificada que deseas permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de errores se aplican únicamente a PyLint los informes de ESLint SA. Para obtener más información sobre los informes de SA, consulte. [Informes de análisis estáticos](#)

Para especificar la gravedad mínima, utilice la Severity propiedad. Para especificar el número máximo de vulnerabilidades, utilice la Number propiedad.

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Errores

StaticAnalysisSecurity

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/
StaticAnalysisSecurity*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/
StaticAnalysisSecurity*

(Opcional)

Especifique el número máximo y la gravedad de las vulnerabilidades de seguridad permitidas en el informe de la SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar las vulnerabilidades de seguridad, debe especificar:

- La gravedad mínima de las vulnerabilidades de seguridad que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo desea HIGH, se HIGH contabilizarán las vulnerabilidades de CRITICAL seguridad.

- El número máximo de vulnerabilidades de seguridad de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de vulnerabilidad de seguridad se aplican únicamente a PyLint los informes de ESLint SA. Para obtener más información sobre los informes de SA, consulte. [Informes de análisis estáticos](#)

Para especificar la gravedad mínima, utilice la Severity propiedad. Para especificar el número máximo de vulnerabilidades, utilice la Number propiedad.

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Vulnerabilidades de seguridad

StaticAnalysisQuality

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/**StaticAnalysisQuality***

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**StaticAnalysisQuality***

(Opcional)

Especifique el número máximo y la gravedad de los problemas de calidad permitidos en el informe de SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar los problemas de calidad, debe especificar:

- La gravedad mínima de los problemas de calidad que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Por ejemplo, si lo desea HIGH, se HIGH contabilizarán los problemas de CRITICAL calidad.

- El número máximo de problemas de calidad de la gravedad especificada que desee permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de calidad solo se aplican a los informes PyLint de ESLint SA. Para obtener más información sobre los informes de SA, consulte [Informes de análisis estáticos](#)

Para especificar la gravedad mínima, utilice la `Severity` propiedad. Para especificar el número máximo de vulnerabilidades, utilice la `Number` propiedad.

Interfaz de usuario correspondiente: pestaña de salida/Informes/Criterios de éxito/Problemas de calidad

StaticAnalysisFinding

*(nombre de la acción) /Outputs/AutoDiscoverReports/SuccessCriteria/
StaticAnalysisFinding*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/
StaticAnalysisFinding*

(Opcional)

Especifique el número máximo y la gravedad de los hallazgos permitidos en el informe de la SA para que el CodeCatalyst informe asociado se marque como aprobado. Para especificar los resultados, debe especificar:

- La gravedad mínima de los hallazgos que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo eligeHIGH, se HIGH contabilizarán CRITICAL los resultados.

- El número máximo de hallazgos de la gravedad especificada que desee permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los resultados se aplican únicamente a los informes de SARIF SA. Para obtener más información sobre los informes de SA, consulte [Informes de análisis estáticos](#).

Para especificar la gravedad mínima, utilice la `Severity` propiedad. Para especificar el número máximo de vulnerabilidades, utilice la `Number` propiedad.

Interfaz de usuario correspondiente: pestaña de salidas/informes/criterios de éxito/hallazgos

Reports

*(nombre de la acción) /Outputs/ **Reports***

(Opcional)

Una sección que especifica la configuración de los informes de pruebas.

Interfaz de usuario correspondiente: pestaña de salidas/informes

nombre-informe-1

(nombre de acción nombre-informe-1) /Outputs/Reports/

(Obligatorio si está incluido) [Reports](#)

El nombre que desea asignar al CodeCatalyst informe que se generará a partir de los informes sin procesar.

Interfaz de usuario correspondiente: pestaña de salidas/informes/configuración manual/nombre del informe

Format

*(nombre /Outputs/Reports/ de **Format**acción, nombre del informe, 1/)*

(Obligatorio si está incluido) [Reports](#)

Especifique el formato de archivo que va a utilizar para sus informes. Los valores posibles son los siguientes.

- Para los informes de prueba:
 - Para Cucumber JSON, especifique Cucumber (editor visual) o CUCUMBERJSON (editor YAML).
 - Para JUnit XML, especifique JUnit (editor visual) o JUNITXML (editor YAML).
 - Para NUnit XML, especifique NUnit (editor visual) o **NUNITXML** (editor YAML).
 - Para NUnit 3 XML, especifique NUnit3 (editor visual) o (editor YAML). NUNIT3XML
 - Para Visual Studio TRX, especifique Visual Studio TRX (editor visual) o (editor YAML). VISUALSTUDIOTRX
 - Para TestNG XML, especifique TestNG (editor visual) o TESTNGXML (editor YAML).

- Para los informes de cobertura de código:
 - Para Clover XML, especifique Clover (editor visual) o CLOVERXML (editor YAML).
 - Para Cobertura XML, especifique Cobertura (editor visual) o COBERTURAXML (editor YAML).
 - Para JaCoCo XML, especifique JaCoCo(editor visual) o JACOCOXML (editor YAML).
 - Para el SimpleCov JSON generado por [simplecov](#), no por [simplecov-json](#), especifique [Simplecov](#) (editor visual) o (editor YAML). SIMPLECOV
- Para los informes de análisis de composición de software (SCA):
 - Para SARIF, especifique SARIF (editor visual) o SARIFSCA (editor YAML).

Interfaz de usuario correspondiente: pestaña de salidas, informes, configuración manual de informes, adición o configuración de informes, nombre del informe 1, tipo y formato del informe

Configuration

(nombre de acción/) **Configuration**

(Obligatorio) Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

Container

(nombre de *la acción /Configuration/ Container*)

(Opcional)

Especifique la imagen de Docker, o el contenedor, que utilizará la acción para completar su procesamiento. Puede especificar una de las [imágenes activas](#) incluidas o puede usar la suya propia. CodeCatalyst Si decide usar su propia imagen, puede residir en Amazon ECR, Docker Hub u otro registro. Si no especificas una imagen de Docker, la acción utilizará una de las imágenes activas para su procesamiento. Para obtener información sobre qué imagen activa se utiliza de forma predeterminada, consulte [Imágenes activas](#).

Para obtener más información sobre cómo especificar su propia imagen de Docker, consulte [Asignación de una imagen de Docker de un entorno de ejecución personalizado a una acción](#).

Interfaz de usuario correspondiente: imagen de Docker del entorno de ejecución (opcional)

Registry

(nombre de la *acción /Configuration/Container/ Registry*)

(Obligatorio si `Container` está incluido)

Especifique el registro en el que se guarda la imagen. Los valores válidos son:

- CODECATALYST(editor YAML)

La imagen se guarda en el CodeCatalyst registro.

- Docker Hub (editor visual) o DockerHub (editor YAML)

La imagen se almacena en el registro de imágenes de Docker Hub.

- Otro registro (editor visual) o Other (editor YAML)

La imagen se guarda en un registro de imágenes personalizado. Se puede utilizar cualquier registro disponible públicamente.

- Amazon Elastic Container Registry (editor visual) o ECR (editor YAML)

La imagen se almacena en un repositorio de imágenes de Amazon Elastic Container Registry. Para usar una imagen en un repositorio de Amazon ECR, esta acción necesita acceso a Amazon ECR. Para habilitar este acceso, debe crear un [rol de IAM](#) que incluya los siguientes permisos y una política de confianza personalizada. (Si lo desea, puede modificar un rol existente para incluir los permisos y la política).

El rol de IAM debe incluir los siguientes permisos en su política de roles:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

El rol de IAM debe incluir la siguiente política de confianza personalizada:

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

```
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codecatalyst-runner.amazonaws.com",
      "codecatalyst.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
]
```

Para obtener más información sobre la creación de funciones de IAM, consulte [Creación de una función mediante políticas de confianza personalizadas \(consola\)](#) en la Guía del usuario de IAM.

Una vez que haya creado el rol, debe asignarlo a la acción a través de un entorno. Para obtener más información, consulte [Asociar un entorno, una conexión de cuenta y un rol de IAM a una acción de flujo de trabajo](#).

Interfaz de usuario correspondiente: Amazon Elastic Container Registry, Docker Hub y otras opciones de registro

Image

(nombre de la *acción* `/Configuration/Container/ Image`)

(Obligatorio si Container está incluido)

Especifique uno de los siguientes valores:

- Si utiliza un CODECATALYST registro, defina la imagen en una de las siguientes [imágenes activas](#):
 - CodeCatalystLinux_x86_64:2024_03
 - CodeCatalystLinux_x86_64:2022_11
 - CodeCatalystLinux_Arm64:2024_03
 - CodeCatalystLinux_Arm64:2022_11
 - CodeCatalystLinuxLambda_x86_64:2024_03
 - CodeCatalystLinuxLambda_x86_64:2022_11

- CodeCatalystLinuxLambda_Arm64:2024_03
- CodeCatalystLinuxLambda_Arm64:2022_11
- CodeCatalystWindows_x86_64:2022_11
- Si utiliza un registro de Docker Hub, defina la imagen con el nombre de la imagen de Docker Hub y la etiqueta opcional.

Ejemplo: postgres:latest

- Si utiliza un registro de Amazon ECR, defina la imagen en el URI del registro de Amazon ECR.

Ejemplo: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo

- Si utiliza un registro personalizado, defina la imagen en el valor esperado por el registro personalizado.

Interfaz de usuario correspondiente: imagen de docker del entorno de ejecución (si el registro es **CODECATALYST**), imagen de Docker Hub (si el registro es Docker Hub), URL de imagen ECR (si el registro es Amazon Elastic Container Registry) y URL de imagen (si el registro es Other registry).

Steps

(nombre de acción) /Configuration/ Steps

(Obligatorio)

Especifique los comandos de shell que desea ejecutar durante la acción para instalar, configurar y ejecutar las herramientas de compilación.

A continuación, se muestra un ejemplo de cómo crear un proyecto de npm:

Steps:

- Run: npm install
- Run: npm run build

A continuación, se muestra un ejemplo de cómo especificar las rutas de los archivos:

Steps:

- Run: cd \$ACTION_BUILD_SOURCE_PATH_WorkflowSource/app && cat file2.txt
- Run: cd \$ACTION_BUILD_SOURCE_PATH_MyBuildArtifact/build-output/ && cat file.txt

Para obtener más información sobre cómo especificar las rutas de los archivos, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña de configuración/comandos de shell

Packages

(nombre de la *acción /Configuration/ Packages*)

(Opcional)

Una sección en la que puedes especificar un repositorio de paquetes que la acción utilizará para resolver las dependencias. Los paquetes permiten almacenar y compartir de forma segura los paquetes de software utilizados para el desarrollo de aplicaciones.

Para obtener más información sobre los paquetes, consulte [Paquetes en CodeCatalyst](#).

Interfaz de usuario correspondiente: pestaña de configuración/paquetes

NpmConfiguration

(nombre de la *acción /Configuration/Packages/ NpmConfiguration*)

(Obligatorio si [Packages](#) está incluido)

Una sección que define la configuración del formato del paquete npm. Una acción utiliza esta configuración durante la ejecución de un flujo de trabajo.

Para obtener más información sobre la configuración del paquete npm, consulte [Con npm](#).

Interfaz de usuario correspondiente: pestaña de configuración/Paquetes/Agregar configuración/ npm

PackageRegistries

*(nombre de la acción) /Configuration/Packages/NpmConfiguration/
PackageRegistries*

(Obligatorio si [Packages](#) está incluido)

Una sección en la que puede definir las propiedades de configuración de una secuencia de repositorios de paquetes.

Interfaz de usuario correspondiente: pestaña de configuración/paquetes/Añadir configuración/npm/
Añadir repositorio de paquetes

PackagesRepository

*(nombre de la acción) /Configuration/Packages/NpmConfiguration/
PackageRegistries/ **PackagesRepository***

(Obligatorio si [Packages](#) está incluido)

Especifique el nombre del repositorio de CodeCatalyst paquetes que desea que utilice la acción.

Si especificas varios repositorios predeterminados, el último repositorio tendrá prioridad.

Para obtener más información sobre los repositorios de paquetes, consulte. [Repositorios de paquetes](#)

Interfaz de usuario correspondiente: pestaña de configuración/Paquetes/Agregar configuración/NPM/
Agregar repositorio de paquetes/Repositorio de paquetes

Scopes

*(nombre de la acción) /Configuration/Packages/NpmConfiguration/
PackageRegistries/ **Scopes***

(Opcional)

Especifique la secuencia de ámbitos que desee definir en el registro de paquetes. Al definir los ámbitos, el repositorio de paquetes especificado se configura como el registro de todos los ámbitos de la lista. Si se solicita un paquete con ese alcance a través del cliente npm, utilizará ese repositorio en lugar del predeterminado. El nombre de cada ámbito debe tener el prefijo «@».

Si incluye ámbitos principales, el último repositorio tendrá prioridad.

Si Scopes se omite, el repositorio de paquetes especificado se configura como el registro predeterminado para todos los paquetes utilizados por la acción.

Para obtener más información sobre los ámbitos, consulte Paquetes [Espacios de nombres de paquetes](#) con [ámbito](#).

Interfaz de usuario correspondiente: pestaña de configuración/Paquetes/Agregar configuración/NPM/
Agregar repositorio de paquetes/Ámbitos: opcional

Referencia de acción «Amazon S3 publish»

La siguiente es la referencia de YAML de la definición de acción para la acción de publicación de Amazon S3. Para obtener información sobre cómo utilizar esta acción, consulte [Añadir la acción «Amazon S3 publish»](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
S3Publish\_nn:
  Identifier: aws/s3-publish@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
    Variables:
      - Name: variable-name-1
        Value: variable-value-1
      - Name: variable-name-2
        Value: variable-value-2
  Environment:
```

```
Name: environment-name  
Connections:  
- Name: account-connection-name  
  Role: iam-role-name  
Configuration:  
SourcePath: my/source  
DestinationBucketName: s3-bucket-name  
TargetPath: my/target
```

S3Publish

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede utilizar comillas para activar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: S3Publish_nn.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de la acción

Identifier

(*S3Publish*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: aws/s3-publish@v1.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ S3Publish_nn/ aws/s3-publish @v1 label

DependsOn

(*S3Publish*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(*S3Publish*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*S3Publish*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/tipo de cálculo

Fleet

(*S3Publish*/Compute/Fleet)

(Opcional)

Especifique la máquina o la flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/flota de cómputo

Timeout

(*S3Publish*/Timeout)

(Obligatorio)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que CodeCatalyst finalice la acción. El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Inputs

(*S3Publish*/Inputs)

(Opcional)

La `Inputs` sección define los datos que se `S3Publish` necesitan durante la ejecución de un flujo de trabajo.

Note

Se permite un máximo de cuatro entradas (una fuente y tres artefactos) para cada acción de AWS CDK despliegue. Las variables no se tienen en cuenta para este total.

Si necesita hacer referencia a archivos que se encuentran en entradas diferentes (por ejemplo, una fuente y un artefacto), la entrada de origen es la entrada principal y el artefacto es la entrada secundaria. Las referencias a los archivos en las entradas secundarias llevan un prefijo especial para distinguirlas de las principales. Para obtener más detalles, consulte [Ejemplo: hacer referencia a archivos en varios artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(`S3Publish/Inputs/Sources`)

(Obligatorio si los archivos que desea publicar en Amazon S3 están almacenados en un repositorio de origen)

Si los archivos que desea publicar en Amazon S3 están almacenados en un repositorio de origen, especifique la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es `esWorkflowSource`.

Si los archivos que desea publicar en Amazon S3 no se encuentran en un repositorio de origen, deben residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(`S3Publish/Inputs/Artifacts`)

(Obligatorio si los archivos que desea publicar en Amazon S3 están almacenados en un [artefacto de salida](#) de una acción anterior)

Si los archivos que desea publicar en Amazon S3 están contenidos en un artefacto generado por una acción anterior, especifique ese artefacto aquí. Si sus archivos no están contenidos en un artefacto, deben residir en su repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Variables - input

(*S3Publish*/Inputs/Variables)

(Opcional)

Especifique una secuencia de pares de nombre/valor que definan las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de entradas/variables: opcional

Environment

(*S3Publish*/Environment)

(Obligatorio)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/Entorno, conexión, rol, entorno

Name

(*S3Publish*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno, conexión, rol, entorno

Connections

(*S3Publish*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajoEnvironment.

Para obtener más información sobre las conexiones de cuentas, consulte[Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte[Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno/conexión/rola'/ Conexión

Name

(*S3Publish*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno, conexión, función, conexión

Role

(*S3Publish*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM que la acción de publicación de Amazon S3 utiliza para acceder a los archivos de Amazon S3 AWS y copiarlos. Asegúrese de que esta función incluya:

- La siguiente política de permisos:

⚠ Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

- La siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

Asegúrese de que este rol esté asociado a la conexión de su cuenta. Para obtener más información sobre cómo asociar un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#)

Note

Si lo desea, puede especificar el nombre del CodeCatalystWorkflowDevelopmentRole-*spaceName* rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno/conexión/rola'/ Función

Configuration

(*S3Publish*/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

SourcePath

(*S3Publish*/Configuration/SourcePath)

(Obligatorio)

Especifique el nombre y la ruta del directorio o archivo que desee publicar en Amazon S3. El directorio o archivo puede residir en un repositorio de origen o en un artefacto de una acción anterior y es relativo al repositorio de origen o a la raíz del artefacto.

Ejemplos:

La especificación `./myFolder/` copia el contenido `/myFolder` de en Amazon S3 y conserva la estructura de directorios subyacente.

Especificar `./myFolder/myfile.txt` copias únicamente `myfile.txt` en Amazon S3. (Se elimina la estructura de directorios).

No puede utilizar caracteres comodín.

Note

Es posible que tenga que añadir un prefijo al directorio o a la ruta del archivo para indicar en qué artefacto o fuente encontrarlo. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña de configuración/ruta de origen

DestinationBucketName

(*S3Publish*/Configuration/DestinationBucketName)

(Obligatorio)

Especifique el nombre del depósito de Amazon S3 en el que desea publicar los archivos.

Interfaz de usuario correspondiente: pestaña de configuración/depósito de destino (opcional)

TargetPath

(*S3Publish*/Configuration/TargetPath)

(Opcional)

Especifique el nombre y la ruta del directorio en Amazon S3 en el que desea publicar los archivos. Si el directorio no existe, se creará. La ruta del directorio no debe incluir el nombre del bucket.

Ejemplos:

`myS3Folder`

```
./myS3Folder/myS3Subfolder
```

Interfaz de usuario correspondiente: pestaña de configuración/directorio de destino: opcional

referencia de acción «AWS CDKbootstrap»

La siguiente es la referencia YAML de la definición de acción para la acción de AWS CDKbootstrap. Para obtener información sobre cómo utilizar esta acción, consulte. [Añadir la acción «AWS CDKbootstrap»](#)

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKBootstrapAction\_nn:
  Identifier: aws/cdk-bootstrap@v1
  DependsOn:
    - action-name
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
  Outputs:
```

Artifacts:

- **Name:** `cdk_bootstrap_artifacts`

Files:

- `"cdk.out/**/*"`

Environment:

Name: `environment-name`

Connections:

- **Name:** `account-connection-name`

Role: `iam-role-name`

Configuration:

Region: `us-west-2`

CdkCliVersion: `version`

CDKBootstrapAction

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: `CDKBootstrapAction_nn`.

Interfaz de usuario correspondiente: pestaña de configuración o nombre para mostrar de la acción

Identifier

(*CDKBootstrapAction*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: `aws/cdk-bootstrap@v1`.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ `CDKBootstrapAction_nn/ aws/cdk-bootstrap @v1` label

DependsOn

(*CDKBootstrapAction*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(*CDKBootstrapAction*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*CDKBootstrapAction*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/tipo de cálculo

Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/Flota de cómputo

Timeout

(*CDKBootstrapAction*/Timeout)

(Obligatorio)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que finalice la acción. CodeCatalyst El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Inputs

(*CDKBootstrapAction*/Inputs)

(Opcional)

La `Inputs` sección define los datos que necesita la acción de AWS CDKarranque durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de entradas

Note

Solo se permite una entrada (ya sea una fuente o un artefacto) para cada acción de AWS CDKarranque.

Sources

(*CDKBootstrapAction*/Inputs/Sources)

(Obligatorio si AWS CDK la aplicación está almacenada en un repositorio de fuentes)

Si tu AWS CDK aplicación está almacenada en un repositorio de origen, especifica la etiqueta de ese repositorio de origen. La acción de AWS CDKarranque sintetiza la aplicación en este repositorio antes de iniciar el proceso de arranque. Actualmente, la única etiqueta de repositorio compatible es `WorkflowSource`

Si tu AWS CDK aplicación no está incluida en un repositorio de origen, debe residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(*CDKBootstrapAction*/Inputs/Artifacts)

(Obligatorio si tu AWS CDK aplicación está almacenada en un [artefacto de salida](#) de una acción anterior)

Si tu AWS CDK aplicación está incluida en un artefacto generado por una acción anterior, especifica ese artefacto aquí. La acción de AWS CDKarranque sintetiza la aplicación del artefacto especificado en una CloudFormation plantilla antes de iniciar el proceso de arranque. Si tu AWS CDK aplicación no está incluida en un artefacto, debe residir en tu repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas/Artefactos (opcional)

Outputs

(*CDKBootstrapAction*/Outputs)

(Opcional)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts - output

(*CDKBootstrapAction*/Outputs/Artifacts)

(Opcional)

Especifique los artefactos generados por la acción. Puedes hacer referencia a estos artefactos como entrada en otras acciones.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de salidas/artefactos

Name

(*CDKBootstrapAction*/Outputs/Artifacts/Name)

(Obligatorio si [Artifacts - output](#) está incluido)

Especifique el nombre del artefacto que contendrá la AWS CloudFormation plantilla sintetizada por la acción de AWS CDKarranque en tiempo de ejecución. El valor predeterminado es `cdk_bootstrap_artifacts`. Si no especificas un artefacto, la acción sintetizará la plantilla, pero no la guardará en un artefacto. Considera la posibilidad de guardar la plantilla sintetizada en un artefacto para conservar un registro de la misma para probarla o solucionar problemas.

Interfaz de usuario correspondiente: pestaña de salidas/Artifectos/Añadir artefacto/Construir nombre de artefacto

Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

([Artifacts - output](#)Obligatorio si está incluido)

Especifique los archivos que desee incluir en el artefacto. Debe especificar si "cdk.out/**/*" desea incluir la AWS CloudFormation plantilla sintetizada de su AWS CDK aplicación.

Note

cdk.out es el directorio predeterminado en el que se guardan los archivos sintetizados. Si especificó un directorio de salida que no sea cdk.out el cdk.json del archivo, especifique ese directorio aquí en lugar de cdk.out.

Interfaz de usuario correspondiente: pestaña de salidas/Artifectos/Agregar artefacto/Archivos producidos por build

Environment

(*CDKBootstrapAction*/Environment)

(Obligatorio)

Especifica el entorno que se va a usar con la acción. CodeCatalyst

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(*CDKBootstrapAction*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*CDKBootstrapAction*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajoEnvironment.

Para obtener más información sobre las conexiones de cuentas, consulte[Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte[Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de cuenta

Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta

Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM que la acción de arranque utiliza para acceder a la pila de arranque y añadirlaAWS CDK. AWS Asegúrese de que esta función incluya las siguientes políticas:

Note

Los permisos que se muestran en la siguiente política de permisos son los que requiere el `cdk bootstrap` comando para realizar su arranque. Estos permisos pueden cambiar si el comando `AWS CDK bootstrap` cambia.

Warning

Utilice este rol únicamente con la acción `AWS CDKbootstrap`. Es muy permisivo y su uso con otras acciones puede suponer un riesgo para la seguridad.

- La siguiente política de permisos:

Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "ssm:GetParameterHistory",
        "ecr:PutImageScanningConfiguration",
        "cloudformation:*",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ssm:GetParameters",
        "iam:PutRolePolicy",
        "ssm:GetParameter",
        "ssm>DeleteParameters",
        "ecr>DeleteRepository",
        "ssm:PutParameter",

```

```

        "ssm:DeleteParameter",
        "iam:PassRole",
        "ecr:SetRepositoryPolicy",
        "ssm:GetParametersByPath",
        "ecr:DescribeRepositories",
        "ecr:GetLifecyclePolicy"
    ],
    "Resource": [
        "arn:aws:ssm:aws-region:aws-account:parameter/cdk-bootstrap/*",
        "arn:aws:cloudformation:aws-region:aws-account:stack/CDKToolkit/*",
        "arn:aws:ecr:aws-region:aws-account:repository/cdk-*",
        "arn:aws:iam::aws-account:role/cdk-*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:RegisterType",
        "cloudformation:CreateUploadBucket",
        "cloudformation:ListExports",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:SetTypeDefaultVersion",
        "cloudformation:RegisterPublisher",
        "cloudformation:ActivateType",
        "cloudformation:ListTypes",
        "cloudformation:DeactivateType",
        "cloudformation:SetTypeConfiguration",
        "cloudformation:DeregisterType",
        "cloudformation:ListTypeRegistrations",
        "cloudformation:EstimateTemplateCost",
        "cloudformation:DescribeAccountLimits",
        "cloudformation:BatchDescribeTypeConfigurations",
        "cloudformation:CreateStackSet",
        "cloudformation:ListStacks",
        "cloudformation:DescribeType",
        "cloudformation:ListImports",
        "s3:*",
        "cloudformation:PublishType",
        "ecr:CreateRepository",
        "cloudformation:DescribePublisher",
        "cloudformation:DescribeTypeRegistration",
        "cloudformation:TestType",
        "cloudformation:ValidateTemplate",
    ]
}

```

```

        "cloudformation:ListTypeVersions"
    ],
    "Resource": "*"
}
]
}

```

Note

La primera vez que utilice el rol, utilice el siguiente comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- La siguiente política de confianza personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Asegúrese de añadir este rol a la conexión de su cuenta. Para obtener más información sobre cómo añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Si lo desea, puede especificar el nombre del CodeCatalystWorkflowDevelopmentRole-*spaceName* rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Te recomendamos que solo utilices este rol en los tutoriales y en situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Rol'/ Rol

Configuration

(*CDKBootstrapAction*/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

Region

(*CDKBootstrapAction*/Configuration/Region)

(Obligatorio)

Especifique el lugar Región de AWS en el que se desplegará la pila de arranque. Esta región debe coincidir con aquella en la que se implementa AWS CDK la aplicación. Para ver una lista de códigos de regiones, consulte [Puntos de conexión regionales](#).

Interfaz de usuario correspondiente: pestaña de configuración/región

CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

(Opcional)

Esta propiedad está disponible en la versión 1.0.13 o posterior de la acción de AWS CDKdespliegue y en la versión 1.0.8 o posterior de la AWS CDK acción de arranque.

Especifique uno de los siguientes valores:

- La versión completa de la interfaz de línea de AWS Cloud Development Kit (AWS CDK) comandos (CLI) (también denominada AWS CDK kit de herramientas) que desea que utilice esta acción. Ejemplo:2.102.1. Considere la posibilidad de especificar una versión completa para garantizar la coherencia y la estabilidad al crear e implementar la aplicación.

Or (Disyunción)

- latest. Considere la posibilidad de especificar latest para aprovechar las funciones y correcciones más recientes de la CLI de CDK.

La acción descargará la versión especificada (o la última versión) de la AWS CDK CLI a la [imagen de CodeCatalyst compilación](#) y, a continuación, utilizará esta versión para ejecutar los comandos necesarios para implementar la aplicación CDK o arrancar el entorno. AWS

Para obtener una lista de las versiones de CDK CLI compatibles que puede utilizar, consulte [AWS CDKVersiones](#).

Si omite esta propiedad, la acción utilizará una versión AWS CDK CLI predeterminada que se describe en uno de los temas siguientes:

- [Versiones de la CLI de CDK utilizadas por la acción «AWS CDK implementar»](#)
- [Versiones de CDK CLI utilizadas por la acción «AWS CDK bootstrap»](#)

Interfaz de usuario correspondiente: pestaña de configuración/versión AWS CDKCLI

referencia de acción «AWS CDK desplegar»

La siguiente es la referencia YAML de la definición de acción para la acción de AWS CDK despliegue. Para obtener información sobre cómo utilizar esta acción, consulte [Añadir la acción «AWS CDK desplegar»](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la

interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKDeploy\_nn:
  Identifier: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
  Outputs:
    Artifacts:
      - Name: cdk_artifact
    Files:
      - "cdk.out/**/*"
  Environment:
    Name: environment-name
    Connections:
      - Name: account-connection-name
        Role: iam-role-name
  Configuration:
    StackName: my-cdk-stack
    Region: us-west-2
    Tags: '{"key1": "value1", "key2": "value2"}'
    Context: '{"key1": "value1", "key2": "value2"}'
    CdkCliVersion: version
    CdkRootPath: directory/cdk.json
```

```
CfnOutputVariables: ["CfnOutputKey1", "CfnOutputKey2", "CfnOutputKey3"]  
CloudAssemblyRootPath: path-to-cdk.out
```

CDKDeploy

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede utilizar comillas para activar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: CDKDeploy_nn.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de la acción

Identifier

(*CDKDeploy*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: aws/cdk-deploy@v1.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ CDKDeploy_nn/ aws/cdk-deploy @v1 label

DependsOn

(*CDKDeploy*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que la acción de despliegue se ejecute. AWS CDK Se recomienda especificar la acción de AWS CDK arranque en la DependsOn propiedad, de la siguiente manera:

```
CDKDeploy:
```

```
Identifier: aws/cdk-deploy@v1
DependsOn:
  - CDKBootstrap
```

Note

El arranque es un requisito previo obligatorio para implementar una aplicación. AWS CDK Si no incluye la acción de AWS CDK Bootstrap en su flujo de trabajo, debe buscar otra forma de implementar la pila de AWS CDK arranque antes de ejecutar la acción de despliegue. AWS CDK Para obtener más información, consulte [Requisitos previos](#) en [Añadir la acción «AWS CDK desplegar»](#).

Para obtener más información sobre la funcionalidad «depende de», consulte. [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(*CDKDeploy*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*CDKDeploy*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/tipo de cálculo

Fleet

(*CDKDeploy*/Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/Flota de cómputo

Timeout

(*CDKDeploy*/Timeout)

(Obligatorio)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que finalice la acción. CodeCatalyst El mínimo es de 5 minutos

y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Inputs

(*CDKDeploy*/Inputs)

(Opcional)

La Inputs sección define los datos que se *CDKDeploy* necesitan durante la ejecución de un flujo de trabajo.

Note

Solo se permite una entrada (una fuente o un artefacto) para cada acción de AWS CDK despliegue.

Interfaz de usuario correspondiente: pestaña Entradas

Sources

(*CDKDeploy*/Inputs/Sources)

(Obligatorio si la AWS CDK aplicación que quieres implementar está almacenada en un repositorio de origen)

Si AWS CDK la aplicación está almacenada en un repositorio de origen, especifique la etiqueta de ese repositorio de origen. La acción de AWS CDK despliegue sintetiza la aplicación en este repositorio antes de iniciar el proceso de despliegue. Actualmente, la única etiqueta admitida es `WorkflowSource`.

Si tu AWS CDK aplicación no está incluida en un repositorio de origen, debe residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(*CDKDeploy*/Inputs/Artifacts)

(Obligatorio si la AWS CDK aplicación que quieres implementar está almacenada en un [artefacto de salida](#) de una acción anterior)

Si tu AWS CDK aplicación está incluida en un artefacto generado por una acción anterior, especifica ese artefacto aquí. La acción de AWS CDK despliegue sintetiza la aplicación del artefacto especificado en una CloudFormation plantilla antes de iniciar el proceso de despliegue. Si tu AWS CDK aplicación no está incluida en un artefacto, debe residir en tu repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas/Artefactos (opcional)

Outputs

(*CDKDeploy*/Outputs)

(Opcional)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts - output

(*CDKDeploy*/Outputs/Artifacts)

(Opcional)

Especifique los artefactos generados por la acción. Puedes hacer referencia a estos artefactos como entrada en otras acciones.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de salidas/artefactos

Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Obligatorio si [Artifacts - output](#) está incluido)

Especifique el nombre del artefacto que contendrá la AWS CloudFormation plantilla sintetizada por la acción de AWS CDK despliegue en tiempo de ejecución. El valor predeterminado es `cdk_artifact`. Si no especificas un artefacto, la acción sintetizará la plantilla pero no la guardará en un artefacto. Considera la posibilidad de guardar la plantilla sintetizada en un artefacto para conservar un registro de la misma para probarla o solucionar problemas.

Interfaz de usuario correspondiente: pestaña de salidas/Artifactos/Añadir artefacto/Construir nombre de artefacto

Files

(*CDKDeploy*/Outputs/Artifacts/Files)

([Artifacts - output](#)Obligatorio si está incluido)

Especifique los archivos que desee incluir en el artefacto. Debe especificar si `"cdk.out/**/*"` desea incluir la AWS CloudFormation plantilla sintetizada de su AWS CDK aplicación.

Note

`cdk.out` es el directorio predeterminado en el que se guardan los archivos sintetizados. Si especificó un directorio de salida que no sea `cdk.out` el `cdk.json` del archivo, especifique ese directorio aquí en lugar de `cdk.out`.

Interfaz de usuario correspondiente: pestaña de salidas/Artifactos/Agregar artefacto/Archivos producidos por build

Environment

(*CDKDeploy*/Environment)

(Obligatorio)

Especifica el entorno que se va a usar con la acción. CodeCatalyst

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(*CDKDeploy*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*CDKDeploy*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo `Environment`.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de cuenta

Name

(*CDKDeploy*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta

Role

(*CDKDeploy*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM que la acción de despliegue utiliza para acceder a la pila de aplicaciones y desplegarla AWS CDK . AWS AWS CDK Asegúrese de que esta función incluya:

- La siguiente política de permisos:

Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackResources"
      ],
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::aws-account:role/cdk-*"
    }
  ]
}
```

- La siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Asegúrese de añadir este rol a la conexión de su cuenta. Para obtener más información sobre cómo añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Si lo desea, puede especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Rol'/ Rol

Configuration

(*CDKDeploy*/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

StackName

(*CDKDeploy*/Configuration/StackName)

(Obligatorio)

El nombre de la pila de AWS CDK aplicaciones, tal como aparece en el archivo de punto de entrada del directorio de la AWS CDK bin aplicación. *En el siguiente ejemplo, se muestra el contenido de un archivo de TypeScript punto de entrada, con el nombre de la pila resaltado en cursiva roja.* Si el archivo de puntos de entrada está en otro idioma, tendrá un aspecto similar.

```
import * as cdk from 'aws-cdk-lib';
import { CdkWorkshopTypescriptStack } from '../lib/cdk_workshop_typescript-stack';

const app = new cdk.App();
new CdkWorkshopTypescriptStack(app, 'CdkWorkshopTypescriptStack');
```

Solo puede especificar una pila.

Tip

Si tiene varias pilas, puede crear una pila principal con pilas anidadas. A continuación, puedes especificar la pila principal en esta acción para desplegar todas las pilas.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de pila

Region

(*CDKDeploy*/Configuration/Region)

(Obligatorio)

Especifique el lugar Región de AWS en el que se desplegará la pila de AWS CDK aplicaciones. Para ver una lista de códigos de regiones, consulte [Puntos de conexión regionales](#).

Interfaz de usuario correspondiente: pestaña/región de configuración

Tags

(*CDKDeploy*/Configuration/Tags)

(Opcional)

Especifique las etiquetas que desee aplicar a AWS los recursos de la pila de AWS CDK aplicaciones. Las etiquetas se aplican a la propia pila, así como a los recursos individuales de la pila. Para obtener más información sobre el etiquetado, consulte [Etiquetado](#) en la Guía para AWS Cloud Development Kit (AWS CDK) desarrolladores.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/ Etiquetas

Context

(*CDKDeploy*/Configuration/Context)

(Opcional)

Especifique los contextos, en forma de pares clave-valor, para asociarlos a la pila de aplicaciones. AWS CDK Para obtener más información sobre los contextos, consulte los [contextos de ejecución](#) en la Guía AWS Cloud Development Kit (AWS CDK) para desarrolladores.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/contexto

CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Opcional)

Esta propiedad está disponible en la versión 1.0.13 o posterior de la acción de AWS CDK despliegue y en la versión 1.0.8 o posterior de la acción de arranque.AWS CDK

Especifique uno de los siguientes valores:

- La versión completa de la interfaz de línea de AWS Cloud Development Kit (AWS CDK) comandos (CLI) (también denominada AWS CDK kit de herramientas) que desea que utilice esta acción. Ejemplo:2.102.1. Considere la posibilidad de especificar una versión completa para garantizar la coherencia y la estabilidad al crear e implementar la aplicación.

Or (Disyunción)

- `latest`. Considere la posibilidad de especificar `latest` para aprovechar las funciones y correcciones más recientes de la CLI de CDK.

La acción descargará la versión especificada (o la última versión) de la AWS CDK CLI a la [imagen de CodeCatalyst compilación](#) y, a continuación, utilizará esta versión para ejecutar los comandos necesarios para implementar la aplicación CDK o arrancar el entorno. AWS

Para obtener una lista de las versiones de CDK CLI compatibles que puede utilizar, consulte [AWS CDK Versiones](#).

Si omite esta propiedad, la acción utilizará una versión AWS CDK CLI predeterminada que se describe en uno de los temas siguientes:

- [Versiones de la CLI de CDK utilizadas por la acción «AWS CDK implementar»](#)
- [Versiones de CDK CLI utilizadas por la acción «AWS CDK bootstrap»](#)

Interfaz de usuario correspondiente: pestaña de configuración/versión AWS CDK CLI

CdkRootPath

(*CDKDeploy*/Configuration/CdkRootPath)

(Opcional)

La ruta al directorio que contiene el `cdk.json` archivo del AWS CDK proyecto. La acción de AWS CDK despliegue se ejecuta desde esta carpeta y cualquier resultado creado por la acción se añadirá a este directorio. Si no se especifica, la acción de AWS CDK despliegue asume que el `cdk.json` archivo se encuentra en la raíz del AWS CDK proyecto.

Interfaz de usuario correspondiente: pestaña de configuración o directorio donde reside el `cdk.json`

CfnOutputVariables

(*CDKDeploy*/Configuration/CfnOutputVariables)

(Opcional)

Especifique qué `CfnOutput` construcciones del código de su AWS CDK aplicación desea exponer como variables de salida del flujo de trabajo. A continuación, puede hacer referencia a las variables

de salida del flujo de trabajo en acciones posteriores del flujo de trabajo. Para obtener más información sobre las variables de CodeCatalyst, consulte [Trabajar con variables](#).

Por ejemplo, si el código de AWS CDK la aplicación tiene este aspecto:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'bucketName', {
      value: bucket.bucketName,
      description: 'The name of the s3 bucket',
      exportName: 'myBucket',
    });
    const table = new dynamodb.Table(this, 'todos-table', {
      partitionKey: { name: 'todoId', type: dynamodb.AttributeType.NUMBER },
      billingMode: dynamodb.BillingMode.PAY_PER_REQUEST,
      removalPolicy: RemovalPolicy.DESTROY,
    })
    new CfnOutput(this, 'tableName', {
      value: table.tableName,
      description: 'The name of the dynamodb table',
      exportName: 'myDynamoDbTable',
    });
    ...
  }
}
```

... y su `CfnOutputVariables` propiedad tiene este aspecto:

```
Configuration:
...
CfnOutputVariables: ["bucketName","tableName"]
```

... entonces la acción genera las siguientes variables de salida del flujo de trabajo:

Clave	Valor
bucketName	bucket.bucketName
tableName	table.tableName

A continuación, puede hacer referencia a `tableName` las variables `bucketName` y en acciones posteriores. Para aprender a hacer referencia a las variables de salida del flujo de trabajo en acciones posteriores, consulte [Hacer referencia a una variable predefinida](#).

Si no especifica ningún `CfnOutput` componente fijo en la `CfnOutputVariables` propiedad, la acción expone las cuatro primeras variables de salida (o menos) que encuentre como variables de CloudFormation salida del flujo de trabajo. Para obtener más información, consulte [«AWS CDKdesplegar» variables de acción](#).

Tip

Para obtener una lista de todas las variables de CloudFormation salida que produce la acción, ejecuta el flujo de trabajo que contiene la acción de AWS CDK despliegue una vez y, a continuación, consulta la pestaña Registros de la acción. Los registros contienen una lista de todas las variables CloudFormation de salida asociadas a tu AWS CDK aplicación. Una vez que sepas cuáles son todas las CloudFormation variables, puedes especificar cuáles quieres convertir en variables de salida del flujo de trabajo mediante la `CfnOutputVariables` propiedad.

Para obtener más información sobre las variables de AWS CloudFormation salida, consulta la documentación de la `CfnOutput` construcción, disponible en la [clase `CfnOutput` \(construcción\)](#) de la referencia de la AWS Cloud Development Kit (AWS CDK) API.

Interfaz de usuario correspondiente: pestaña de configuración/variables AWS CloudFormation de salida

CloudAssemblyRootPath

(*CDKDeploy*/Configuration/CloudAssemblyRootPath)

(Opcional)

Si ya sintetizó la pila de su AWS CDK aplicación en un ensamblaje en la nube (mediante la `cdk synth` operación), especifique la ruta raíz del directorio del ensamblaje en la nube (`cdk.out`). La AWS CloudFormation plantilla ubicada en el directorio de ensamblaje en la nube especificado se implementará mediante la acción de AWS CDK implementación Cuenta de AWS mediante el `cdk deploy --app` comando. Cuando la `--app` opción está presente, la `cdk synth` operación no se lleva a cabo.

Si no especifica un directorio de ensamblaje en la nube, la acción de AWS CDK despliegue ejecutará el `cdk deploy` comando sin la `--app` opción. Sin `--app` esta opción, la `cdk deploy` operación sintetizará (`cdk synth`) e implementará tu AWS CDK aplicación en tu Cuenta de AWS.

¿Por qué especificaría un ensamblaje de nube sintetizado y existente cuando la acción de «AWS CDK despliegue» puede realizar la síntesis en tiempo de ejecución?

Es posible que desee especificar un ensamblaje de nube sintetizado existente para:

- Asegúrese de implementar exactamente el mismo conjunto de recursos cada vez que se ejecute la acción de «AWS CDK despliegue»

Si no especificas un ensamblaje en la nube, es posible que la acción de AWS CDK despliegue sintetice e implemente diferentes archivos en función de cuándo se ejecute. Por ejemplo, la acción de AWS CDK despliegue podría sintetizar un ensamblaje en la nube con un conjunto de dependencias durante una fase de prueba y otro conjunto de dependencias durante una fase de producción (si esas dependencias cambiaran de una etapa a otra). Para garantizar una paridad exacta entre lo que se ha probado y lo que se ha desplegado, se recomienda sintetizar una vez y, a continuación, utilizar el campo Ruta del directorio del ensamblaje de nube (editor visual) o la `CloudAssemblyRootPath` propiedad (editor YAML) para especificar el ensamblaje de nube ya sintetizado.

- Utilice herramientas y gestores de paquetes no estándares con la aplicación AWS CDK

Durante una `synth` operación, la acción de AWS CDK despliegue intenta ejecutar la aplicación con herramientas estándar, como `npm` o `pip`. Si la acción no puede ejecutar correctamente tu aplicación con esas herramientas, la síntesis no se producirá y la acción fallará. Para solucionar este problema, puedes especificar los comandos exactos necesarios para ejecutar la aplicación correctamente en el `cdk.json` archivo de la AWS CDK aplicación y, a continuación, sintetizarla mediante un método que no implique la acción de AWS CDK implementación. Una vez generado el ensamblaje de nube, puedes especificarlo en el campo Ruta al directorio del ensamblaje en la nube (editor visual) o en la `CloudAssemblyRootPath` propiedad (editor YAML) de la acción de AWS CDK despliegue.

Para obtener información sobre cómo configurar el `cdk.json` archivo para que incluya comandos para instalar y ejecutar la AWS CDK aplicación, consulte [Especificar el comando de la aplicación](#).

Para obtener información sobre los `cdk synth` comandos `cdk deploy` y, así como sobre la `--app` opción, consulte [Implementar pilas, Sintetizar pilas](#) y [omitir la síntesis](#) en la Guía para desarrolladores.AWS Cloud Development Kit (AWS CDK)

Para obtener información sobre Cloud Assembly, consulta [Cloud Assembly](#) en la referencia de la API.AWS Cloud Development Kit (AWS CDK)

Interfaz de usuario correspondiente: pestaña de configuración o ruta al directorio de ensamblaje en la nube

referencia de acción «AWS Lambda invocar»

La siguiente es la referencia YAML de la definición de acción para la acción de AWS Lambda invocación. Para obtener información sobre cómo utilizar esta acción, consulte. [Añadir la acción «AWS Lambdainvocar»](#)

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
LambdaInvoke\_nn:
  Identifier: aws/lambda-invoke@v1
  DependsOn:
    - dependent-action
  Compute:
    Type: EC2 | Lambda
```

```

Fleet: fleet-name
Timeout: timeout-minutes
Inputs:
  # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
  Artifacts:
    - request-payload
  Variables:
    - Name: variable-name-1
      Value: variable-value-1
    - Name: variable-name-2
      Value: variable-value-2
Environment:
  Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
Configuration:
  Function: my-function/function-arn
  AWSRegion: us-west-2
  # Specify RequestPayload or RequestPayloadFile, but not both.
  RequestPayload: '{"firstname": "Li", lastname: "Jean", "company": "ExampleCo",
"team": "Development"}'
  RequestPayloadFile: my/request-payload.json
  ContinueOnError: true/false
  LogType: Tail/None
  ResponseFilters: '{"name": ".name", "company": ".department.company"}'
Outputs:
  Artifacts:
    - Name: lambda_artifacts
      Files:
        - "lambda-response.json"

```

LambdaInvoke

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: `Lambda_Invoke_Action_Workflow_nn`.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de la acción

Identifier

(*LambdaInvoke*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: `aws/lambda-invoke@v1`.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ `LambdaInvoke_nn/ aws/lambda-invoke @v1` label

DependsOn

(*LambdaInvoke*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de (opcional)

Compute

(*LambdaInvoke*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*LambdaInvoke*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/tipo de cálculo

Fleet

(*LambdaInvoke*/Compute/Fleet)

(Opcional)

Especifique la máquina o la flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/flota de cómputo

Timeout

(*LambdaInvoke*/Timeout)

(Obligatorio)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que CodeCatalyst finalice la acción. El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Inputs

(*LambdaInvoke*/Inputs)

(Obligatorio)

La Inputs sección define los datos que necesita la acción de AWS Lambda invocación durante la ejecución de un flujo de trabajo.

Note

Solo se permite una entrada (una fuente o un artefacto) por acción de AWS Lambda invocación. Las variables no se tienen en cuenta para este total.

Interfaz de usuario correspondiente: pestaña de entradas

Sources


(*LambdaInvoke*/Inputs/Sources)

(Obligatorio si [RequestPayloadFile](#) se proporciona)

Si quieres pasar un archivo JSON de carga útil de una solicitud a la acción de AWS Lambda invocación y este archivo de carga útil está almacenado en un repositorio de origen, especifica la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es `WorkflowSource`

Si el archivo de carga útil de la solicitud no está contenido en un repositorio de origen, debe residir en un artefacto generado por otra acción.

Para obtener más información sobre el archivo de carga útil, consulte. [RequestPayloadFile](#)

 Note

En lugar de especificar un archivo de carga útil, puedes añadir el código JSON de la carga útil directamente a la acción mediante la propiedad. RequestPayload Para obtener más información, consulte [RequestPayload](#).

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)


Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Obligatorio si [RequestPayloadFile](#) se proporciona)

Si quieres pasar un archivo JSON de carga útil de una solicitud a la acción de AWS Lambda invocación y este archivo de carga está contenido en un [artefacto de salida de una acción anterior](#), [especifica ese artefacto](#) aquí.

Para obtener más información sobre el archivo de carga útil, consulte. [RequestPayloadFile](#)

 Note

En lugar de especificar un archivo de carga útil, puedes añadir el código JSON de la carga útil directamente a la acción mediante la propiedad. RequestPayload Para obtener más información, consulte [RequestPayload](#).

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte. [Trabajar con artefactos](#)

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Opcional)

Especifique una secuencia de pares de nombre/valor que definan las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de entradas/variables: opcional

Environment

(*LambdaInvoke*/Environment)

(Obligatorio)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(*LambdaInvoke*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Conexión/Role'/ Entorno

Connections

(*LambdaInvoke*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo `Environment`.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno/conexión/rola'/ Conexión

Name

(*LambdaInvoke*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno, conexión, función, conexión

Role

(*LambdaInvoke*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM que la acción de AWS Lambda invocación utiliza para acceder a la función AWS Lambda e invocarla. Asegúrese de que esta función incluya:

- La siguiente política de permisos:

Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:aws-account:function:function-name"
    }
  ]
}
```



```

    }
  ]
}

```

- La siguiente política de confianza personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Asegúrese de que este rol esté asociado a la conexión de su cuenta. Para obtener más información sobre cómo asociar un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#)

Note

Si lo desea, puede especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Te recomendamos que solo utilices este rol en los tutoriales y en situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/'Entorno/conexión/rola'/ Función

Configuration

(*LambdaInvoke*/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

Function

(*LambdaInvoke*/Configuration/Function)

(Obligatorio)

Especifique la AWS Lambda función que invocará esta acción. Puede especificar el nombre de la función o su nombre de recurso de Amazon (ARN). Puede encontrar el nombre o el ARN en la consola de Lambda.

Note

La AWS cuenta en la que reside la función Lambda puede ser diferente de la cuenta especificada en. `Connections` :

Interfaz de usuario correspondiente: pestaña/función de configuración

AWSRegion

(*LambdaInvoke*/Configuration/AWSRegion)

(Obligatorio)

Especifique la AWS región en la que reside su AWS Lambda función. Para obtener una lista de los códigos de región, consulte los [puntos finales regionales](#) en el Referencia general de AWS.

Interfaz de usuario correspondiente: pestaña de configuración/depósito de destino (opcional)

RequestPayload

(*LambdaInvoke*/Configuration/RequestPayload)

(Opcional)

Si quieres pasar la carga útil de una solicitud a la acción de AWS Lambda invocación, especifica la carga útil de la solicitud aquí, en formato JSON.

Ejemplo de carga útil de solicitud:

```
'{ "key": "value" }'
```

Si no desea pasar una carga útil de solicitud a la función Lambda, omite esta propiedad.

Note

Puede especificar `RequestPayload` o `RequestPayloadFile`, pero no ambos.

Para obtener más información sobre la carga útil de la solicitud, consulte el tema [Invoke en la referencia](#) de la API.AWS Lambda

Interfaz de usuario correspondiente: pestaña de configuración/carga útil de solicitud (opcional)

RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Opcional)

Si desea pasar la carga útil de una solicitud a la acción de AWS Lambda invocación, especifique aquí la ruta a este archivo de carga útil de la solicitud. El archivo debe estar en formato JSON.

El archivo de carga útil de la solicitud puede residir en un repositorio de origen o en un artefacto de una acción anterior. La ruta del archivo es relativa al repositorio de origen o a la raíz del artefacto.

Si no desea pasar una carga útil de solicitud a la función Lambda, omite esta propiedad.

Note

Puede especificar `RequestPayload` o `RequestPayloadFile`, pero no ambos.

Para obtener más información sobre el archivo de carga útil de la solicitud, consulte el tema [Invoke en la referencia](#) de la API.AWS Lambda

Interfaz de usuario correspondiente: pestaña de configuración/archivo de carga útil de solicitud (opcional)

ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Opcional)

Especifique si desea marcar la acción de AWS Lambda invocación como realizada correctamente incluso si la función invocada AWS Lambda falla. Considere configurar esta propiedad `true` para permitir que se inicien las acciones posteriores del flujo de trabajo a pesar del error de Lambda.

El valor predeterminado es no realizar la acción si la función Lambda falla («desactivada» en el editor visual o `false` en el editor YAML).

Interfaz de usuario correspondiente: pestaña de configuración/Continuar en caso de error

LogType

(*LambdaInvoke*/Configuration/LogType)

(Opcional)

Especifique si desea incluir los registros de errores en la respuesta de la función Lambda después de invocarla. Puede ver estos registros en la pestaña Registros de la acción de invocación de Lambda en la consola. CodeCatalyst Los valores posibles son los siguientes:

- `Tail`— devuelve los registros
- `None`— no devuelva los registros

El valor predeterminado es `Tail`.

Para obtener más información sobre el tipo de registro, consulta el tema [Invoke](#) en la referencia de la AWS Lambda API.

Para obtener más información acerca de cómo visualizar los archivos de registro, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Interfaz de usuario correspondiente: pestaña de configuración/tipo de registro

ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Opcional)

Especifique qué claves de la carga útil de respuesta de Lambda desea convertir en variables de salida. A continuación, puede hacer referencia a las variables de salida en acciones posteriores de su flujo de trabajo. Para obtener más información sobre las variables de CodeCatalyst, consulte [Trabajar con variables](#).

Por ejemplo, si la carga útil de su respuesta tiene este aspecto:

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

... y tus filtros de respuesta tienen este aspecto:

```
Configuration:
...
ResponseFilters: '{"name": ".name", "company": ".department.company"}'
```

... entonces la acción genera las siguientes variables de salida:

Clave	Valor
name	Saanvi
company	Amazon

A continuación, puede hacer referencia a company las variables name y en acciones posteriores.

Si no especifica ninguna `claveResponseFilters`, la acción convierte cada clave de nivel superior de la respuesta de Lambda en una variable de salida. Para obtener más información, consulte [«AWS Lambda invocar» variables de acción](#).

Considere la posibilidad de utilizar filtros de respuesta para limitar las variables de salida generadas solo a las que realmente desee utilizar.

Interfaz de usuario correspondiente: pestaña de configuración/filtros de respuesta (opcionales)

Outputs

(*LambdaInvoke*/Outputs)

(Opcional)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts

(*LambdaInvoke*/Outputs/Artifacts)

(Opcional)

Especifique los artefactos generados por la acción. Puedes hacer referencia a estos artefactos como entrada en otras acciones.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de salidas/Artifacts/ Nombre del artefacto de construcción

Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Opcional)

Especifique el nombre del artefacto que contendrá la carga útil de respuesta de Lambda que devuelve la función Lambda. El valor predeterminado es `lambda_artifacts`. Si no especifica un artefacto, la carga útil de la respuesta de Lambda se puede ver en los registros de la acción, que

están disponibles en la pestaña Registros de la acción en la consola. CodeCatalyst Para obtener más información acerca de cómo visualizar los archivos de registro, consulte [Ver el estado y los detalles de la ejecución del flujo de trabajo](#).

Interfaz de usuario correspondiente: pestaña de salidas/Artifectos/ Nombre del artefacto de construcción

Files

(*LambdaInvoke*/Outputs/Artifacts/Files)

(Opcional)

Especifique los archivos que desee incluir en el artefacto. Debe especificarlo `lambda-response.json` para que se incluya el archivo de carga útil de respuesta de Lambda.

Interfaz de usuario correspondiente: pestaña de salidas, artefactos y archivos producidos por la compilación

Referencia de acción «Implementar AWS CloudFormation pila»

La siguiente es la referencia de YAML de la definición de acciones para la acción Implementar AWS CloudFormation pila. Para obtener información sobre cómo utilizar esta acción, consulte [Añadir la acción «Desplegar AWS CloudFormation pila»](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
```

DeployCloudFormationStack:Identifier: *aws/cfn-deploy@v1*DependsOn:

- *build-action*

Compute:Type: *EC2 | Lambda*Fleet: *fleet-name*Timeout: *timeout-minutes*Environment:Name: *environment-name*Connections:

- Name: *account-connection-name*

Role: *DeployRole*Inputs:Sources:

- *source-name-1*

Artifacts:

- *CloudFormation-artifact*

Configuration:name: *stack-name*region: *us-west-2*template: *template-path*role-arn: *arn:aws:iam::123456789012:role/StackRole*capabilities: *CAPABILITY_IAM,CAPABILITY_NAMED_IAM,CAPABILITY_AUTO_EXPAND*parameter-overrides: *KeyOne=ValueOne,KeyTwo=ValueTwo | path-to-JSON-file*no-execute-changeset: *1|0*fail-on-empty-changeset: *1|0*disable-rollback: *1|0*termination-protection: *1|0*timeout-in-minutes: *minutes*notification-arns: *arn:aws:sns:us-east-1:123456789012:MyTopic,arn:aws:sns:us-**east-1:123456789012:MyOtherTopic*monitor-alarm-arns: *arn:aws:cloudwatch::123456789012:alarm/**MyAlarm,arn:aws:cloudwatch::123456789012:alarm/MyOtherAlarm*monitor-timeout-in-minutes: *minutes*tags: *'[{"Key":"MyKey1","Value":"MyValue1"}, {"Key":"MyKey2","Value":"MyValue2"}]'*

DeployCloudFormationStack

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9),

guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: `DeployCloudFormationStack_nn`.

Interfaz de usuario correspondiente: pestaña de configuración o nombre para mostrar de la acción

Identifier

(DeployCloudFormationStack/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: `aws/cfn-deploy@v1`.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ `DeployCloudFormationStack_nn/`
`aws/cfn-deploy @v1` label

DependsOn

(DeployCloudFormationStack/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte. [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(DeployCloudFormationStack/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(DeployCloudFormationStack/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/tipo de cálculo

Fleet

(DeployCloudFormationStack/Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará su flujo de trabajo o sus acciones. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/Flota de cómputo

Timeout

(DeployCloudFormationStack/Timeout)

(Opcional)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que finalice la acción. CodeCatalyst El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/Tiempo de espera en minutos (opcional)

Environment

(DeployCloudFormationStack/Environment)

(Obligatorio)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(DeployCloudFormationStack/Environment/Name)

[Environment](#) Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*DeployCloudFormationStack*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajoEnvironment.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de cuenta

Name

(*DeployCloudFormationStack*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta

Role


(*DeployCloudFormationStack*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM que la acción de implementación de la pila utiliza para acceder al servicio. AWS CloudFormation AWS AWS CloudFormation


Asegúrese de que esta función incluya las siguientes políticas:

- La siguiente política de permisos:

 Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo para la seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

 Note

La primera vez que utilice el rol, utilice el siguiente comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- La siguiente política de confianza personalizada:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

Asegúrese de añadir este rol a la conexión de la cuenta en su espacio. Para obtener más información sobre cómo añadir un rol de IAM a una conexión de cuenta, consulta [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Si lo desea, puede especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.


Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Rol'/ Rol

Inputs

(*DeployCloudFormationStack*/Inputs)

(Opcional)

La Inputs sección define los datos que se necesitan durante la ejecución de un flujo de trabajo. `DeployCloudFormationStack`

 Note

Se permite un máximo de cuatro entradas (una fuente y tres artefactos) por cada acción de Deploy AWS CloudFormation stack.

Si necesitas hacer referencia a archivos que se encuentran en entradas diferentes (por ejemplo, una fuente y un artefacto), la entrada de origen es la entrada principal y el artefacto es la entrada secundaria. Las referencias a los archivos en las entradas secundarias llevan un prefijo especial para distinguirlas de las principales. Para obtener más detalles, consulte [Ejemplo: hacer referencia a archivos en varios artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(`DeployCloudFormationStack/Inputs/Sources`)

(Necesaria si tu AWS SAM plantilla CloudFormation o plantilla está almacenada en un repositorio de fuentes)

Si su AWS SAM plantilla CloudFormation o plantilla está almacenada en un repositorio de origen, especifique la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es `WorkflowSource`.

Si su AWS SAM plantilla CloudFormation o su plantilla no se encuentran en un repositorio de origen, deben residir en un artefacto generado por otra acción o en un bucket de Amazon S3.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(`DeployCloudFormationStack/Inputs/Artifacts`)

(Obligatorio si tu AWS SAM plantilla CloudFormation o plantilla está almacenada en un [artefacto de salida](#) de una acción anterior)

Si la AWS SAM plantilla CloudFormation o que desea implementar está incluida en un artefacto generado por una acción anterior, especifique ese artefacto aquí. Si la CloudFormation plantilla no está incluida en un artefacto, debe residir en el repositorio de origen o en un bucket de Amazon S3.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Configuration

(DeployCloudFormationStack/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

name

(DeployCloudFormationStack/Configuration/name)

(Obligatorio)

Especifique un nombre para la CloudFormation pila que la acción Implementar AWS CloudFormation pila crea o actualiza.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de pila

region

(DeployCloudFormationStack/Configuration/region)

(Obligatorio)

Especifique el lugar Región de AWS en el que se desplegará la pila. Para ver una lista de códigos de regiones, consulte [Puntos de conexión regionales](#).

Interfaz de usuario correspondiente: pestaña de configuración/región de pila

template

(DeployCloudFormationStack/Configuration/template)

(Obligatorio)

Especifique el nombre y la ruta de su archivo CloudFormation o AWS SAM de plantilla. La plantilla puede estar en formato JSON o YAML y puede residir en un repositorio de origen, un artefacto de una acción anterior o un bucket de Amazon S3. Si el archivo de plantilla está en un repositorio o artefacto fuente, la ruta es relativa a la fuente o raíz del artefacto. Si la plantilla está en un bucket de Amazon S3, la ruta es el valor de la URL del objeto de la plantilla.

Ejemplos:

```
./MyFolder/MyTemplate.json
```

```
MyFolder/MyTemplate.yml
```

```
https://MyBucket.s3.us-west-2.amazonaws.com/MyTemplate.yml
```

Note

Es posible que tengas que añadir un prefijo a la ruta del archivo de la plantilla para indicar en qué artefacto o fuente puedes encontrarla. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña/plantilla de configuración

role-arn

(DeployCloudFormationStack/Configuration/role-arn)


(Obligatorio)

Especifique el nombre de recurso de Amazon (ARN) del rol de pila. CloudFormation utiliza este rol para acceder a los recursos de la pila y modificarlos. Por ejemplo:
`arn:aws:iam::123456789012:role/StackRole.`

Asegúrese de que la función de pila incluya:

- Una o más políticas de permisos. Las políticas dependen de los recursos de los que dispongas. Por ejemplo, si su pila incluye una AWS Lambda función, debe añadir permisos que permitan el acceso a Lambda. Si ha seguido el tutorial que se describe en [Tutorial: Implemente una aplicación](#)

[sin servidor mediante AWS CloudFormation](#), incluye un procedimiento denominado [Para crear un rol de pila](#) que enumera los permisos que necesita la función de pila si va a implementar una pila de aplicaciones sin servidor típica.


 Warning

Limite los permisos a los necesarios por el CloudFormation servicio para acceder a los recursos de su pila. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

- La siguiente política de confianza:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Si lo desea, asocie este rol a la conexión de su cuenta. Para obtener más información sobre cómo asociar un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#). Si no asocia el rol de pila a la conexión de la cuenta, el rol de pila no aparecerá en la lista desplegable del rol de pila del editor visual; sin embargo, el ARN del rol aún se puede especificar en el `role-arn` campo mediante el editor YAML.

 Note

Si lo deseas, puedes especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos

muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/función de pila (opcional)

capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Obligatorio)

Especifique una lista de las capacidades de IAM necesarias para poder AWS CloudFormation crear determinadas pilas. En la mayoría de los casos, puede quedarse capabilities con el valor predeterminado de. CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_AUTO_EXPAND

Si ves `##[error] requires capabilities: [capability-name]` en los registros de tus acciones de Deploy AWS CloudFormation stack, consulta [¿Cómo puedo corregir los errores de las capacidades de IAM?](#) para obtener información sobre cómo solucionar el problema.

Para obtener más información sobre las capacidades de IAM, consulte [Reconocimiento de los recursos de IAM en AWS CloudFormation las plantillas](#) de la Guía del usuario de IAM.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Capacidades

parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Opcional)

Especifique los parámetros en su AWS CloudFormation AWS SAM plantilla que no tengan valores predeterminados o para los que desee especificar valores no predeterminados. Para obtener más información sobre los parámetros, consulte [Parámetros](#) en la Guía del AWS CloudFormation usuario.

El parameter-overrides establecimiento acepta:

- Un archivo JSON que contiene los parámetros y valores.
- Una lista de parámetros y valores separados por comas.

Para especificar un archivo JSON

1. Asegúrese de que el archivo JSON utilice una de las siguientes sintaxis:

```
{
  "Parameters": {
    "Param1": "Value1",
    "Param2": "Value2",
    ...
  }
}
```

O bien...

```
[
  {
    "ParameterKey": "Param1",
    "ParameterValue": "Value1"
  },
  ...
]
```

(Existen otras sintaxis, pero no las admiten CodeCatalyst en el momento de escribir este artículo). Para obtener más información sobre cómo especificar CloudFormation los parámetros en un archivo JSON, consulte la [Sintaxis JSON admitida](#) en la Referencia de AWS CLI comandos.

2. Especifique la ruta al archivo JSON mediante uno de los siguientes formatos:

- Si el archivo JSON reside en un artefacto de salida de una acción anterior, utilice:

```
file:///artifacts/current-action-name/output-artifact-name/path-to-json-file
```

Consulte el ejemplo 1 para obtener más información.

- Si el archivo JSON reside en el repositorio de origen, usa:

```
file:///sources/WorkflowSource/path-to-json-file
```

Consulte el ejemplo 2 para obtener más información.

Ejemplo 1: El archivo JSON reside en un artefacto de salida

```
##My workflow YAML
...
Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ParamArtifact
          Files:
            - params.json
    Configuration:
      ...
  MyDeployCFNStackAction:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      parameter-overrides: file:///artifacts/MyDeployCFNStackAction/
ParamArtifact/params.json
```

Ejemplo 2: El archivo JSON reside en el repositorio de origen, en una carpeta llamada my/folder

```
##My workflow YAML
...
Actions:
  MyDeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      parameter-overrides: file:///sources/WorkflowSource/my/folder/params.json
```

Para usar una lista de parámetros separados por comas

- Añada pares de nombre-valor de los parámetros a la `parameter-overrides` propiedad con el siguiente formato:

param-1=value-1,param-2=value-2

Por ejemplo, suponiendo la siguiente plantilla: AWS CloudFormation

```
##My CloudFormation template

Description: My AWS CloudFormation template

Parameters:
  InstanceType:
    Description: Defines the Amazon EC2 compute for the production server.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t2.small
      - t3.medium

Resources:
...
```

... puede establecer la `parameter-overrides` propiedad de la siguiente manera:

```
##My workflow YAML
...
Actions:
...
  DeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      parameter-overrides: InstanceType=t3.medium,UseVPC=true
```

Note

Puede especificar un nombre de parámetro sin el valor correspondiente utilizando `undefined` como valor. Por ejemplo:

```
parameter-overrides: MyParameter=undefined
```

El efecto es que, durante una actualización de la pila, CloudFormation utiliza el valor del parámetro existente para el nombre del parámetro dado.

Interfaz de usuario correspondiente:

- Pestaña de configuración/Avanzado/Modificaciones de parámetros
- Pestaña de configuración/Avanzado/Modificaciones de parámetros/ Especifique las anulaciones mediante un archivo
- Pestaña de configuración/Avanzado/Modificaciones de parámetros/ Especifique las anulaciones mediante un conjunto de valores

no-execute-changeset

(*DeployCloudFormationStack*/Configuration/no-execute-changeset)

(Opcional)

Especifique si desea crear el conjunto de cambios y, CodeCatalyst a continuación, detenerlo antes de ejecutarlo CloudFormation . Esto le da la oportunidad de revisar el conjunto de cambios en la CloudFormation consola. Si determina que el conjunto de cambios se ve bien, desactive esta opción y, a continuación, vuelva a ejecutar el flujo de trabajo para CodeCatalyst poder crear y ejecutar el conjunto de cambios sin detenerlo. De forma predeterminada, se crea y ejecuta el conjunto de cambios sin detenerlo. Para obtener más información, consulte el parámetro de AWS CloudFormation [despliegue](#) en la Referencia de AWS CLI comandos. Para obtener más información sobre la visualización de los conjuntos de cambios, consulte [Visualización de un conjunto de cambios](#) en la Guía del AWS CloudFormation usuario.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Conjunto de cambios sin ejecutar

fail-on-empty-changeset

(*DeployCloudFormationStack*/Configuration/fail-on-empty-changeset)

(Opcional)

Especifique si desea que no se ejecute la acción CodeCatalyst Implementar AWS CloudFormation pila si el conjunto de cambios está vacío. CloudFormation (Si un conjunto de cambios está vacío, significa que no se realizaron cambios en la pila durante la última implementación). La opción predeterminada es permitir que la acción continúe si el conjunto de cambios está vacío y devolver un UPDATE_COMPLETE mensaje aunque la pila no se haya actualizado.

Para obtener más información sobre esta configuración, consulte el parámetro de AWS CloudFormation [despliegue](#) en la Referencia de AWS CLI comandos. Para obtener más información sobre los conjuntos de cambios, consulte [Actualización de pilas mediante conjuntos de cambios](#) en la Guía del AWS CloudFormation usuario.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Error en el conjunto de cambios vacío

disable-rollback

(DeployCloudFormationStack/Configuration/disable-rollback)

(Opcional)

Especifique si desea revertir la implementación de la pila CodeCatalyst en caso de que se produzca un error. La reversión devuelve la pila al último estado estable conocido. La opción predeterminada es habilitar las reversiones. Para obtener más información sobre esta configuración, consulte el parámetro de AWS CloudFormation [despliegue](#) en la Referencia de AWS CLI comandos.

Para obtener más información sobre cómo la acción Implementar AWS CloudFormation pila gestiona las reversiones, consulte [Configuración de reversiones](#).

Para obtener más información sobre cómo revertir una pila, consulte [las opciones de error de apilamiento](#) en la Guía del AWS CloudFormation usuario.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Desactivar la reversión

termination-protection

(DeployCloudFormationStack/Configuration/termination-protection)

(Opcional)

Especifique si desea que la pila de implementación añada una protección de terminación a la AWS CloudFormation pila que está desplegando. Si un usuario intenta eliminar una pila con protección de terminación habilitada, la eliminación falla y la pila, así como su estado, no cambian. La opción predeterminada es deshabilitar la protección de terminación. Para obtener más información, consulte [Proteger una pila para que no se elimine](#) en la Guía del AWS CloudFormation usuario.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Protección de terminación

timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/timeout-in-minutes)

(Opcional)

Especifique la cantidad de tiempo, en minutos, que CloudFormation debe transcurrir antes de agotar el tiempo de espera de las operaciones de creación de la pila y establecer el estado de la pila en `CREATE_FAILED`. Si CloudFormation no puede crear toda la pila en el tiempo asignado, se produce un error de creación porque el tiempo de espera se agota y la pila vuelve a su estado anterior.

De forma predeterminada, no hay tiempo de espera para la creación de pilas. Sin embargo, puede que determinados recursos individuales tengan sus propios tiempos de espera en función de la naturaleza del servicio que implementan. Por ejemplo, si se agota el tiempo de espera de un recurso individual de la pila, también se agota el tiempo de espera de creación de la pila aunque todavía no se haya agotado el tiempo de espera especificado para la creación de la pila.

Interfaz de usuario correspondiente: pestaña de configuración/avanzada/tiempo de espera
CloudFormation

notification-arns

(*DeployCloudFormationStack*/Configuration/notification-arns)

(Opcional)

Especifique el ARN de un tema de Amazon SNS al que CodeCatalyst desee enviar los mensajes de notificación. Por ejemplo, `arn:aws:sns:us-east-1:111222333:MyTopic`. Cuando se ejecute la acción Implementar una AWS CloudFormation pila, CodeCatalyst coordina con ella CloudFormation para enviar una notificación por cada AWS CloudFormation evento que se produzca durante el proceso de creación o actualización de la pila. (Los eventos están visibles en la pestaña Eventos de la AWS CloudFormation consola para la pila). Puede especificar hasta cinco temas. Para obtener más información, consulte [¿Qué es Amazon SNS?](#)


Interfaz de usuario correspondiente: pestaña de configuración/avanzada/ARN de notificación

monitor-alarm-arns

(*DeployCloudFormationStack*/Configuration/monitor-alarm-arns)

(Opcional)

Especifique el nombre de recurso de Amazon (ARN) de una CloudWatch alarma de Amazon para utilizarla como activador de reversión. Por ejemplo, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Puede tener un máximo de cinco activadores de reversión.

 Note

Si especificas un CloudWatch ARN de alarma, también tendrás que configurar permisos adicionales para permitir el acceso de la acción. CloudWatch Para obtener más información, consulte [Configuración de reversiones](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Monitorear los ARN de alarma

monitor-timeout-in-minutes

(DeployCloudFormationStack/Configuration/monitor-timeout-in-minutes)

(Opcional)

Especifique un período de tiempo, de 0 a 180 minutos, durante el cual se supervisarán las alarmas especificadas. CloudFormation La supervisión comienza después de que se hayan desplegado todos los recursos de la pila. Si la alarma se produce dentro del tiempo de supervisión especificado, la implementación falla y CloudFormation revierte toda la operación de apilamiento.

Predeterminado: 0. CloudFormation solo monitorea las alarmas mientras se despliegan los recursos de la pila, no después.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Tiempo de monitoreo

tags

(DeployCloudFormationStack/Configuration/tags)

(Opcional)

Especifique las etiquetas que desee adjuntar a su pila. CloudFormation Las etiquetas son pares de valores clave arbitrarios que puede utilizar para identificar su pila con fines como la asignación de costes. Para obtener más información sobre qué son las etiquetas y cómo se pueden utilizar, consulte [Etiquetar los recursos](#) en la Guía del usuario de Amazon EC2 para instancias de Linux.

Para obtener más información sobre el etiquetado CloudFormation, consulte [Configuración de las opciones de AWS CloudFormation pila](#) en la Guía del AWS CloudFormation usuario.

Una clave puede tener caracteres alfanuméricos o espacios, y puede tener hasta 127 caracteres. Un valor puede tener caracteres alfanuméricos o espacios y puede tener un máximo de 255 caracteres.

Puede agregar hasta 50 etiquetas únicas para cada pila.

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada/Etiquetas

Referencia de acción «Implementar en Amazon ECS»

La siguiente es la referencia de YAML de definición de acción para la acción Implementar en Amazon ECS. Para obtener información sobre cómo utilizar esta acción, consulte [Añadir la acción «Implementar en Amazon ECS»](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSDeployAction\_nn:
  Identifier: aws/ecs-deploy@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
```

Connections:

- Name: *account-connection-name*
- Role: *DeployToECS*

Inputs:

Specify a source or an artifact, but not both.

Sources:

- *source-name-1*

Artifacts:

- *task-definition-artifact*

Configuration:

region: *us-east-1*
cluster: *ecs-cluster*
service: *ecs-service*
task-definition: *task-definition-path*
force-new-deployment: *false|true*
codedeploy-appspec: *app-spec-file-path*
codedeploy-application: *application-name*
codedeploy-deployment-group: *deployment-group-name*
codedeploy-deployment-description: *deployment-description*

ECSDeployAction

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede utilizar comillas para activar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: ECSDeployAction_nn.

Interfaz de usuario correspondiente: pestaña de configuración o nombre para mostrar de la acción

Identifier

(*ECSDeployAction*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: aws/ecs-deploy@v1.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/etiqueta ECS DeployAction _nn/
aws/ecs-deploy @v1

DependsOn

(*ECSDeployAction*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(*ECSDeployAction*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*ECSDeployAction*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/tipo de cálculo

Fleet

(*ECSDeployAction*/Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará su flujo de trabajo o sus acciones. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/Flota de cómputo

Timeout

(*ECSDeployAction*/Timeout)

(Opcional)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que finalice la acción. CodeCatalyst El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Environment

(*ECSDeployAction*/Environment)

(Obligatorio)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(*ECSDeployAction*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*ECSDeployAction*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo `Environment`.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de cuenta

Name

(*ECSDeployAction*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta

Role

(*ECSDeployAction*/Environment/Connections/Role)

(Obligatorio)

Especifique el nombre de la función de IAM a la que accede AWS la acción Deploy to Amazon ECS. Asegúrese de que esta función incluya las siguientes políticas:

- La siguiente política de permisos:

Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
```



```

    "sns:ListTopics",
    "s3:GetObject",
    "s3:GetObjectVersion",
    "codedeploy:CreateApplication",
    "codedeploy:CreateDeployment",
    "codedeploy:CreateDeploymentGroup",
    "codedeploy:GetApplication",
    "codedeploy:GetDeployment",
    "codedeploy:GetDeploymentGroup",
    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

Note

La primera vez que utilice el rol, utilice el siguiente comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- La siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Asegúrese de añadir este rol a la conexión de su cuenta. Para obtener más información sobre cómo añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Si lo desea, puede especificar el nombre del CodeCatalystWorkflowDevelopmentRole-*spaceName* rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol para su cuenta y su espacio](#). Ten en cuenta que el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol tiene permisos

muy amplios, lo que puede suponer un riesgo para la seguridad. Le recomendamos que utilice esta función únicamente en los tutoriales y situaciones en las que la seguridad sea un problema menor.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Rol'/ Rol

Inputs

(*ECSDeployAction*/Inputs)

(Opcional)

La Inputs sección define los datos que se necesitan durante la ejecución de un flujo de trabajo. *ECSDeployAction*

Note

Solo se permite una entrada (ya sea una fuente o un artefacto) por cada acción de implementación en Amazon ECS.

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(*ECSDeployAction*/Inputs/Sources)

(Necesario si el archivo de definición de tareas está almacenado en un repositorio de origen)

Si el archivo de definición de tareas está almacenado en un repositorio de origen, especifique la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es *WorkflowSource*.

Si el archivo de definición de tareas no está contenido en un repositorio de origen, debe residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(*ECSDeployAction*/Inputs/Artifacts)

(Obligatorio si el archivo de definición de tareas está almacenado en un [artefacto de salida](#) de una acción anterior)

Si el archivo de definición de tareas que desea implementar está contenido en un artefacto generado por una acción anterior, especifique ese artefacto aquí. Si el archivo de definición de tareas no está contenido en un artefacto, debe residir en el repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Configuration

(ECSDeployAction/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

region

(Configuration/region)

(Obligatorio)

Especifique la AWS región en la que residen el clúster y el servicio de Amazon ECS. Para obtener una lista de los códigos de región, consulte los [puntos de enlace regionales](#) en Referencia general de AWS

Interfaz de usuario correspondiente: pestaña de configuración/región

cluster

(ECSDeployAction/Configuration/cluster)

(Obligatorio)

Especifique el nombre de un clúster de Amazon ECS existente. La acción Implementar en Amazon ECS desplegará la aplicación contenerizada como una tarea en este clúster. Para obtener más

información sobre los clústeres de Amazon ECS, consulte [Clústeres](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/clúster

service

(*ECSDeployAction*/Configuration/service)

(Obligatorio)

Especifique el nombre de un servicio Amazon ECS existente que instanciará el archivo de definición de tareas. Este servicio debe residir en el clúster especificado en el `cluster` campo. Para obtener más información sobre los servicios de Amazon ECS, consulte los [servicios de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/servicio

task-definition

(*ECSDeployAction*/Configuration/task-definition)

(Obligatorio)

Especifique la ruta a un archivo de definición de tareas existente. Si el archivo reside en el repositorio de origen, la ruta es relativa a la carpeta raíz del repositorio de origen. Si el archivo reside en un artefacto de una acción de flujo de trabajo anterior, la ruta es relativa a la carpeta raíz del artefacto. Para obtener más información sobre los archivos de definición de tareas, consulte [Definiciones de tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/definición de tareas

force-new-deployment

(*ECSDeployAction*/Configuration/force-new-deployment)

(Obligatorio)

Si está habilitado, el servicio Amazon ECS puede iniciar nuevas implementaciones sin cambios en la definición del servicio. Al forzar una implementación, el servicio detiene todas las tareas que se están ejecutando actualmente y lanza nuevas tareas. Para obtener más información sobre cómo forzar nuevas implementaciones, consulte [Actualización de un servicio](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Predeterminado: false


Interfaz de usuario correspondiente: pestaña de configuración/Forzar una nueva implementación del servicio

codedeploy-appspec


(*ECSDeployAction*/Configuration/codedeploy-appspec)

(Obligatorio si ha configurado su servicio Amazon ECS para utilizar implementaciones azul/verde; de lo contrario, omita)

Especifique el nombre y la ruta de un archivo de especificación de la CodeDeploy aplicación () existente. AppSpec Este archivo debe residir en la raíz del repositorio de CodeCatalyst origen. Para obtener más información sobre AppSpec los archivos, consulte los [archivos de especificación de la CodeDeploy aplicación \(AppSpec\)](#) en la Guía del AWS CodeDeploy usuario.

 Note

Proporcione CodeDeploy información únicamente si ha configurado su servicio Amazon ECS para realizar despliegues azules o verdes. En el caso de las implementaciones de actualizaciones sucesivas (opción predeterminada), omita la información. CodeDeploy Para obtener más información sobre las implementaciones de Amazon ECS, consulte los [tipos de implementación de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

 Note

Es posible que los CodeDeploy campos estén ocultos en el editor visual. Para hacer que aparezcan, consulte [¿Por qué faltan CodeDeploy campos en el editor visual?](#)

Interfaz de usuario correspondiente: pestaña de configuración/ CodeDeploy AppSpec

codedeploy-application

(*ECSDeployAction*/Configuration/codedeploy-application)

(Obligatorio si codedeploy-appspec está incluido)

Especifique el nombre de una CodeDeploy aplicación existente. Para obtener más información sobre CodeDeploy las aplicaciones, consulte [Trabajar con aplicaciones CodeDeploy en](#) la Guía del AWS CodeDeploy usuario.

Interfaz de usuario correspondiente: pestaña de configuración/aplicación CodeDeploy

codedeploy-deployment-group

(*ECSDeployAction*/Configuration/codedeploy-deployment-group)

(Obligatorio si codedeploy-appspec está incluido)

Especifique el nombre de un grupo de CodeDeploy implementación existente. Para obtener más información sobre los grupos de CodeDeploy despliegue, consulte [Trabajar con grupos de despliegue CodeDeploy en](#) la Guía del AWS CodeDeploy usuario.

Interfaz de usuario correspondiente: pestaña de configuración/grupo CodeDeploy de despliegue

codedeploy-deployment-description

(*ECSDeployAction*/Configuration/codedeploy-deployment-description)

(Opcional)

Especifique una descripción de la implementación que creará esta acción. Para obtener más información, consulte [Trabajar con despliegues CodeDeploy en](#) la Guía del AWS CodeDeploy usuario.

Interfaz de usuario correspondiente: pestaña de configuración/descripción CodeDeploy de la implementación

Referencia de acción «Implementar en un clúster de Kubernetes»

A continuación, se muestra la referencia YAML de la definición de acciones para la acción Implementar en el clúster de Kubernetes. Para obtener información sobre cómo utilizar esta acción, consulte. [Añadir la acción «Implementar en un clúster de Kubernetes»](#)

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la

interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
DeployToKubernetesCluster\_nn:
  Identifier: aws/kubernetes-deploy@v1
  DependsOn:
    - build-action
  Compute:
    - Type: EC2 | Lambda
    - Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
    Role: DeployToEKS
  Inputs:
    # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
  Artifacts:
    - manifest-artifact
  Configuration:
    Namespace: namespace
    Region: us-east-1
    Cluster: eks-cluster
    Manifests: manifest-path
```

DeployToKubernetesCluster

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: `DeployToKubernetesCluster_nn`.

Interfaz de usuario correspondiente: pestaña de configuración o nombre para mostrar de la acción

Identifier

(DeployToKubernetesCluster/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: `aws/kubernetes-deploy@v1`.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ `DeployToKubernetesCluster_nn/`
`aws/kubernetes-deploy @v1` label

DependsOn

(DeployToKubernetesCluster/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte. [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(DeployToKubernetesCluster/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*DeployToKubernetesCluster*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/tipo de cálculo

Fleet

(*DeployToKubernetesCluster*/Compute/Fleet)

(Opcional)

Especifique la máquina o flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo

demanda:Linux.x86-64.Large,. Linux.x86-64.XLarge Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. Linux.x86-64.Large

Interfaz de usuario correspondiente: pestaña de configuración/Avanzada: opcional/Flota de cómputo

Timeout

(*DeployToKubernetesCluster*/Timeout)

(Opcional)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que finalice la acción. CodeCatalyst El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Environment

(*DeployToKubernetesCluster*/Environment)

(Obligatorio)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Name

(*DeployToKubernetesCluster*/Environment/Name)

([Environment](#)Obligatorio si está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*DeployToKubernetesCluster*/Environment/Connections)

([Environment](#)Obligatorio si está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajoEnvironment.

Para obtener más información sobre las conexiones de cuentas, consulte[Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte[Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de cuenta

Name

(*DeployToKubernetesCluster*/Environment/Connections/Name)

(Obligatorio)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta


Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Obligatorio)


Especifique el nombre de la función de IAM a la que se accede mediante la acción de implementación en el clúster de Kubernetes. AWS Asegúrese de que esta función incluya las siguientes políticas:

- La siguiente política de permisos:

 Warning

Limite los permisos a los que se muestran en la siguiente política. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

La primera vez que utilice el rol, utilice el siguiente comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

- La siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [

```

```
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
    ],
    },
    "Action": "sts:AssumeRole"
}
]
```

Asegúrese de añadir este rol a:

- La conexión de tu cuenta. Para obtener más información sobre cómo añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).
- Tu Kubernetes ConfigMap. Para obtener más información sobre cómo añadir un rol de IAM a un ConfigMap, consulte [Administrar usuarios y roles de IAM](#) en la documentación. `eksctl`

Tip

Consulte también las [Tutorial: Implementación de una aplicación en Amazon EKS](#) instrucciones sobre cómo añadir un rol de IAM a una conexión de cuenta y. ConfigMap

Note

Si lo desea, puede especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol para su cuenta y su espacio](#). Ten en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Te recomendamos que solo utilices este rol en los tutoriales y en situaciones en las que la seguridad sea un problema menor. Si decide utilizar el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol, asegúrese de añadirlo al ConfigMap archivo siguiendo las instrucciones de la `eksctl` documentación sobre la [gestión de usuarios y funciones de IAM](#).

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/role/' rol

Inputs

(DeployToKubernetesCluster/Inputs)

(Opcional)

La Inputs sección define los datos que se necesitan durante la ejecución de un flujo de trabajo. `DeployToKubernetesCluster`

Note

Solo se permite una entrada (ya sea una fuente o un artefacto) por cada acción de implementación en Amazon EKS.

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(DeployToKubernetesCluster/Inputs/Sources)

(Obligatorio si el archivo de manifiesto está almacenado en un repositorio de origen)

Si tu archivo o archivos de manifiesto de Kubernetes están almacenados en un repositorio de origen, especifica la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es.

`WorkflowSource`

Si los archivos de manifiesto no están contenidos en un repositorio de origen, deben residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(DeployToKubernetesCluster/Inputs/Artifacts)

(Obligatorio si el archivo de manifiesto está almacenado en un [artefacto de salida](#) de una acción anterior)

Si el archivo o los archivos de manifiesto de Kubernetes están contenidos en un artefacto generado por una acción anterior, especifique ese artefacto aquí. Si tus archivos de manifiesto no están contenidos en un artefacto, deben residir en tu repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Configuration

(DeployToKubernetesCluster/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

Namespace

(DeployToKubernetesCluster/Configuration/Namespace)

(Opcional)

Especifique el espacio de nombres de Kubernetes en el que se desplegará la aplicación de Kubernetes. Úselo de `default` si no utiliza espacios de nombres con su clúster. Para obtener más información sobre los espacios de nombres, consulta [Subdividir el clúster mediante espacios de nombres de Kubernetes en la documentación de Kubernetes](#).

Si omite el espacio de nombres, se usará el valor de `default`.

Interfaz de usuario correspondiente: pestaña de configuración/espacio de nombres

Region

(DeployToKubernetesCluster/Configuration/Region)

(Obligatorio)

Especifique la AWS región en la que residen el clúster y el servicio de Amazon EKS. Para obtener una lista de los códigos de región, consulte los [puntos de enlace regionales](#) en Referencia general de AWS.

Interfaz de usuario correspondiente: pestaña de configuración/región

Cluster

(DeployToKubernetesCluster/Configuration/Cluster)

(Obligatorio)

Especifique el nombre de un clúster de Amazon EKS existente. La acción Implementar en un clúster de Kubernetes desplegará la aplicación contenerizada en este clúster. Para obtener más información sobre los clústeres de Amazon EKS, consulte [Clústeres](#) en la Guía del usuario de Amazon EKS.

Interfaz de usuario correspondiente: pestaña de configuración/clúster

Manifests

(DeployToKubernetesCluster/Configuration/Manifests)

(Obligatorio)

Especifique la ruta a sus archivos de manifiesto de Kubernetes con formato YAML, que se denominan archivos de configuración, archivos de configuración o, simplemente, configuraciones, en la documentación de Kubernetes.

Si utilizas varios archivos de manifiesto, colócalos en una sola carpeta y haz referencia a esa carpeta. Kubernetes procesa los archivos de manifiesto de forma alfanumérica, así que asegúrate de anteponer números o letras crecientes a los nombres de los archivos para controlar el orden de procesamiento. Por ejemplo:

```
00-namespace.yaml
```

```
01-deployment.yaml
```

Si los archivos de manifiesto se encuentran en el repositorio de origen, la ruta es relativa a la carpeta raíz del repositorio de origen. Si los archivos residen en un artefacto de una acción de flujo de trabajo anterior, la ruta es relativa a la carpeta raíz del artefacto.

Ejemplos:

```
Manifests/
```

```
deployment.yaml
```

```
my-deployment.yml
```

No utilice caracteres comodín (). *

Note

No se [admiten los gráficos de Helm](#) ni [los archivos de personalización](#).

Para obtener más información sobre los archivos de manifiesto, consulte [Organizar las configuraciones de los recursos](#) en la documentación de Kubernetes.

Interfaz de usuario correspondiente: pestaña de configuración/manifiestos

Referencia de acción «GitHub Acciones»

La siguiente es la referencia YAML de la definición de acción para la acción GitHubAcciones.

Elige una propiedad YAML en el código siguiente para ver su descripción.

Note

La mayoría de las propiedades YAML siguientes tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/github-actions-runner@v1
  DependsOn:
    - dependent-action-name-1
  Compute:
    Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
```

Connections:

- Name: *account-connection-name*
- Role: *iam-role-name*

Inputs:Sources:

- *source-name-1*
- *source-name-2*

Artifacts:

- *artifact-name*

Variables:

- Name: *variable-name-1*
Value: *variable-value-1*
- Name: *variable-name-2*
Value: *variable-value-2*

Outputs:Artifacts:

- Name: *output-artifact-1*

Files:

- *github-output/artifact-1.jar*
- *"github-output/build*"*

- Name: *output-artifact-2*

Files:

- *github-output/artifact-2.1.jar*
- *github-output/artifact-2.2.jar*

Variables:

- *variable-name-1*
- *variable-name-2*

AutoDiscoverReports:

Enabled: *true | false*

ReportNamePrefix: *AutoDiscovered*

IncludePaths:

- *"**/*"*

ExcludePaths:

- *node_modules/cdk/junit.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

Reports:

report-name-1:

Format: *format*

```

IncludePaths:
- "*.xml"
ExcludePaths:
- report2.xml
- report3.xml
SuccessCriteria:
PassRate: percent
LineCoverage: percent
BranchCoverage: percent
Vulnerabilities:
  Severity: CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL
  Number: whole-number

```

Configuration

Steps:

```
- github-actions-code
```

nombre de acción

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de acción

Identifier

*(nombre de acción/) **Identifier***

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

aws/github-actions-runner@v1 Utilícela para GitHub acciones y acciones.

Interfaz de usuario correspondiente: diagrama de flujo de *trabajo/nombre-acción/aws/* @v1
label github-actions-runner

DependsOn

*(**DependsOn**nombre-acción/)*

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de (opcional)

Compute

(nombre de *acción*/) Compute

(Opcional)

El motor informático que se utiliza para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Fleet

(nombre de *la acción /Compute/ Fleet*)

(Opcional)

Especifique la máquina o la flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte [Propiedades de la flota aprovisionada](#)

Si `Fleet` se omite, el valor predeterminado es `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/flota de cómputo (opcional)

Timeout

(nombre de *acción*/) Timeout

(Opcional)

Especifique la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que CodeCatalyst finalice la acción. El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Environment

(nombre de *acción*/) Environment

(Opcional)

Especifique el CodeCatalyst entorno que se va a utilizar con la acción.

Para obtener más información sobre los entornos, consulte [Trabajo con entornos](#) y [Creación de un entorno](#).

Interfaz de usuario correspondiente: pestaña de configuración/entorno/cuenta/rol

Name

(*/Environment/Name* nombre de la acción)

(Obligatorio si [Environment](#) está incluido)

Especifique el nombre de un entorno existente que desee asociar a la acción.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Role'/ Entorno

Connections

(*/Environment/Connections* nombre de la acción)

(Obligatorio si [Environment](#) está incluido)

Especifique la conexión de cuenta que desee asociar a la acción. Puede especificar un máximo de una conexión de cuenta por debajo `Environment`.

Para obtener más información sobre las conexiones de cuentas, consulte [Cuentas de AWS Administrar un espacio](#). Para obtener información sobre cómo asociar una conexión de cuenta a su entorno, consulte [Creación de un entorno](#).

Interfaz de usuario correspondiente: ninguna

Name

(nombre de *la acción /Environment/Connections/ Name*)

(Opcional)

Especifique el nombre de la conexión de la cuenta.

Interfaz de usuario correspondiente: pestaña de configuración/'entorno/cuenta/rola'/ conexión AWS de la cuenta

Role

(*nombre de la acción*) /Environment/Connections/ **Role**

(Opcional)

Especifique el nombre de la función de IAM que utiliza esta acción para acceder y operar en AWS servicios como Amazon S3 y Amazon ECR. Asegúrese de añadir esta función a la conexión de su cuenta. Para añadir un rol de IAM a una conexión de cuenta, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

Note

Es posible que pueda especificar el nombre del `CodeCatalystWorkflowDevelopmentRole-spaceName` rol aquí, siempre que tenga permisos suficientes. Para obtener más información acerca de este rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol para su cuenta y su espacio](#). Tenga en cuenta que el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol tiene permisos muy amplios, lo que puede suponer un riesgo para la seguridad. Te recomendamos que solo utilices este rol en los tutoriales y en situaciones en las que la seguridad sea un problema menor.

⚠ Warning

Limite los permisos a los requeridos por la GitHub acción Action. El uso de un rol con permisos más amplios puede suponer un riesgo de seguridad.

Interfaz de usuario correspondiente: pestaña de configuración/'Ambiente/Cuenta/Rol'/ Rol

Inputs

(**Inputs** nombre-acción/)

(Opcional)

La Inputs sección define los datos que necesita una acción durante la ejecución de un flujo de trabajo.

ℹ Note

Se permite un máximo de cuatro entradas (una fuente y tres artefactos) por acción de GitHub Actions. Las variables no se tienen en cuenta para este total.

Si necesita hacer referencia a archivos que se encuentran en entradas diferentes (por ejemplo, una fuente y un artefacto), la entrada de origen es la entrada principal y el artefacto es la entrada secundaria. Las referencias a los archivos en las entradas secundarias llevan un prefijo especial para distinguirlos de las principales. Para obtener más detalles, consulte [Ejemplo: hacer referencia a archivos en varios artefactos](#).

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(nombre de *la acción /Inputs/ Sources*)

(Opcional)

Especifique las etiquetas que representan los repositorios de origen que necesitará la acción. Actualmente, la única etiqueta admitida es `WorkflowSource` la que representa el repositorio de origen en el que se almacena el archivo de definición de flujo de trabajo.

Si omite una fuente, debe especificar al menos un artefacto de entrada en ella. *action-name/Inputs/Artifacts*

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes: opcional

Artifacts - input

(nombre de la *acción /Inputs/ Artifacts*)

(Opcional)

Especifique los artefactos de acciones anteriores que desee proporcionar como entrada a esta acción. Estos artefactos ya deben estar definidos como artefactos de salida en acciones anteriores.

Si no especifica ningún artefacto de entrada, debe especificar al menos un repositorio de origen en *action-name/Inputs/Sources*.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Note

Si la lista desplegable Artefactos (opcional) no está disponible (editor visual) o si se producen errores al validar el YAML (editor de YAML), es posible que la acción solo admita una entrada. En este caso, intenta eliminar la entrada de origen.

Interfaz de usuario correspondiente: pestaña de entradas/Artefactos (opcional)

Variables - input

(nombre de la *acción /Inputs/ Variables*)

(Opcional)

Especifique una secuencia de pares de nombre/valor que definan las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de entradas/variables: opcional

Outputs

(nombre de *acción*/) Outputs

(Opcional)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts - output

(nombre de *la acción /Outputs/ Artifacts*)

(Opcional)

Especifique el nombre de un artefacto generado por la acción. Los nombres de los artefactos deben ser únicos dentro de un flujo de trabajo y están limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) y guiones bajos (_). No se permiten espacios, guiones (-) ni otros caracteres especiales. No puede utilizar comillas para activar los espacios, guiones y otros caracteres especiales en los nombres de los artefactos de salida.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte. [Trabajar con artefactos](#)

Interfaz de usuario correspondiente: pestaña de salidas/artefactos

Name

(nombre de la *acción /Outputs/Artifacts/ Name*)

(Obligatorio si [Artifacts - output](#) está incluido)

Especifique el nombre de un artefacto generado por la acción. Los nombres de los artefactos deben ser únicos dentro de un flujo de trabajo y están limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) y guiones bajos (_). No se permiten espacios, guiones (-) ni otros caracteres especiales. No puede

utilizar comillas para activar los espacios, guiones y otros caracteres especiales en los nombres de los artefactos de salida.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#)

Interfaz de usuario correspondiente: pestaña de salidas/Artifactos/Añadir artefacto/Construir nombre de artefacto

Files

(nombre de la acción) /Outputs/Artifacts/ Files

(Obligatorio si [Artifacts - output](#) está incluido)


Especifique los archivos que se CodeCatalyst incluyen en el artefacto que se genera con la acción. Estos archivos los genera la acción del flujo de trabajo cuando se ejecuta y también están disponibles en el repositorio de origen. Las rutas de los archivos pueden residir en un repositorio de origen o en un artefacto de una acción anterior, y son relativas al repositorio de origen o a la raíz del artefacto. Puede utilizar patrones globales para especificar las rutas. Ejemplos:

- Para especificar un único archivo que esté en la raíz de la ubicación de compilación o de la ubicación del repositorio de origen, utilice `my-file.jar`.
- Para especificar un único archivo en un subdirectorio, use `directory/my-file.jar` o `directory/subdirectory/my-file.jar`.
- Para especificar todos los archivos, utilice `**/*`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.
- Para especificar todos los archivos y directorios de un directorio denominado `directory`, utilice `directory/**/*`. El patrón glob de `**` indica que debe coincidir con cualquier número de subdirectorios.
- Para especificar todos los archivos de un directorio denominado `directory`, pero no ninguno de sus subdirectorios, utilice `directory/*`.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otro carácter especial, escriba la ruta entre comillas dobles (). "" Para obtener más información sobre los caracteres especiales, consulte [Pautas y convenciones de sintaxis](#)

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

 Note

Puede que tenga que añadir un prefijo a la ruta del archivo para indicar en qué artefacto o fuente puede encontrarlo. Para obtener más información, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña de salidas/Artefactos/Agregar artefacto/Archivos producidos por build

Variables - output

(nombre de la acción) /Outputs/ Variables

(Opcional)

Especifique las variables que desea que exporte la acción para que estén disponibles para su uso en acciones posteriores.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de salidas/Variables/ Añadir variable

nombre de la variable-1

(nombre-de-acción-nombre-de-variable-1) /Outputs/Variables

(Opcional)

Especifique el nombre de la variable que desee que exporte la acción. Esta variable ya debe estar definida en la Steps sección Inputs o de la misma acción.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de salidas/Variables/Añadir variable/ Nombre

AutoDiscoverReports

*(nombre de la acción) /Outputs/ **AutoDiscoverReports***

(Opcional)

Define la configuración de la función de detección automática.

Al habilitar la detección automática, CodeCatalyst busca todos los archivos `Inputs` transferidos a la acción, así como todos los archivos generados por la propia acción, buscando informes de pruebas, cobertura de código y análisis de composición de software (SCA). Para cada informe que se encuentre, lo CodeCatalyst transforma en un CodeCatalyst informe. Un CodeCatalyst informe es un informe que está totalmente integrado en el CodeCatalyst servicio y se puede ver y manipular a través de la CodeCatalyst consola.

Note

De forma predeterminada, la función de detección automática inspecciona todos los archivos. Puede limitar los archivos que se inspeccionan mediante las propiedades [IncludePaths](#) o [ExcludePaths](#).

Interfaz de usuario correspondiente: ninguna

Enabled

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **Enabled***

(Opcional)

Activa o desactiva la función de detección automática.

Los valores válidos son `true` o `false`.

Si `Enabled` se omite, el valor predeterminado es `true`

Interfaz de usuario correspondiente: pestaña de salidas/Informes/Detección automática de informes

ReportNamePrefix

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **ReportNamePrefix***

(Obligatorio si [AutoDiscoverReports](#) está incluido y activado)

Especifique un prefijo que CodeCatalyst preceda a todos los informes que encuentre para asignar un nombre a los informes asociados. CodeCatalyst Por ejemplo, si especifica un prefijo de AutoDiscovered y CodeCatalyst descubre automáticamente dos informes de pruebas TestSuiteOne.xml y TestSuiteTwo.xml, los CodeCatalyst informes asociados se AutoDiscoveredTestSuiteOne denominarán y. AutoDiscoveredTestSuiteTwo

Interfaz de usuario correspondiente: pestaña de salidas/Informes/Detección automática de informes/
Prefijo de informe

IncludePaths

*(nombre de la acción) /Outputs/AutoDiscoverReports/ **IncludePaths***

Or (Disyunción)

*(nombre-acción-nombre-de-informe-1//Outputs/Reports/) **IncludePaths***

(Obligatorio si [AutoDiscoverReports](#) está incluido y activado, o si está incluido) [Reports](#)

Especifique los archivos y las rutas de archivo que se CodeCatalyst incluyen al buscar informes sin procesar. Por ejemplo, si lo especificas "/test/report/*", CodeCatalyst busca en el /test/report/* directorio toda la [imagen de compilación](#) utilizada por la acción. Cuando encuentra ese directorio, CodeCatalyst busca los informes en ese directorio.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otros caracteres especiales, escriba la ruta entre comillas dobles ("). "" Para obtener más información sobre los caracteres especiales, consulte. [Pautas y convenciones de sintaxis](#)

Si se omite esta propiedad, será la predeterminada "**/*", lo que significa que la búsqueda incluirá todos los archivos de todas las rutas.

Note

En el caso de los informes configurados manualmente, IncludePaths debe ser un patrón global que coincida con un único archivo.

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/'Incluir/Excluir rutas'/Incluir rutas
- Pestaña de salidas/Informes/Configurar informes manualmente/ nombre-informe-1 /'Include/Exclude ***rutas'***/ ***Incluir rutas***

ExcludePaths

(/Outputs/AutoDiscoverReports/ExcludePathsnombre-acción)

Or (Disyunción)

(nombre-acción-nombre-de-informe-1//Outputs/Reports/) ExcludePaths

(Opcional)

Especifique los archivos y las rutas de archivo que se CodeCatalyst excluyen al buscar informes sin procesar. Por ejemplo, si lo especifica `/test/my-reports/**/*`, no CodeCatalyst buscará archivos en el `/test/my-reports/` directorio. Para ignorar todos los archivos de un directorio, utilice el patrón `**/*` global.

Note

Si la ruta del archivo incluye uno o más asteriscos (*) u otros caracteres especiales, escriba la ruta entre comillas dobles (). "" Para obtener más información sobre los caracteres especiales, consulte. [Pautas y convenciones de sintaxis](#)

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/'Incluir/Excluir rutas'/Excluir rutas
- Pestaña de salidas/Informes/Configurar informes manualmente/ nombre-informe-1 /'Include/Exclude ***rutas'***/ ***Excluir rutas***

SuccessCriteria

(/Outputs/AutoDiscoverReports/SuccessCriterianombre-acción)

Or (Disyunción)

*(nombre-acción-nombre-de-informe-1//Outputs/Reports/) **SuccessCriteria***

(Opcional)

Especifique los criterios de éxito de los informes de prueba, cobertura del código, análisis de composición de software (SCA) y análisis estático (SA).

Para obtener más información, consulte [Configurar los criterios de éxito de los informes](#).

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Criterios de éxito
- ***Pestaña de salidas, informes, configuración manual de los informes, nombre del informe 1, criterios de éxito***

PassRate

*(nombre /Outputs/AutoDiscoverReports/SuccessCriteria/ **PassRate**de acción)*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**PassRate***

(Opcional)

Especifique el porcentaje de pruebas en un informe de prueba que deben aprobarse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen números decimales. Por ejemplo: 50, 60.5. Los criterios de porcentaje de aprobados se aplican únicamente a los informes de las pruebas. Para obtener más información sobre los informes de las pruebas, consulte [Informes de pruebas](#).

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Criterios de éxito/Tasa de aprobación
- ***Pestaña de resultados, informes, configuración manual de los informes, nombre del informe 1, criterios de éxito, tasa de aprobación***

LineCoverage

*(nombre /Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage** de la acción)*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**LineCoverage***

(Opcional)

Especifique el porcentaje de líneas de un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de línea se aplican únicamente a los informes de cobertura de código. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Criterios de éxito/Cobertura de línea
- ***Pestaña de salidas/Informes/Configuración manual de informes/nombre-de-informe-1 /Criterios de éxito/Cobertura de línea***

BranchCoverage

*(nombre /Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage** de la acción)*

Or (Disyunción)

*(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/**BranchCoverage***

(Opcional)

Especifique el porcentaje de ramificaciones de un informe de cobertura de código que deben cubrirse para que el CodeCatalyst informe asociado se marque como aprobado. Los valores válidos incluyen los números decimales. Por ejemplo: 50, 60.5. Los criterios de cobertura de las sucursales se

aplican únicamente a los informes de cobertura de códigos. Para obtener más información sobre los informes de cobertura de código, consulte [Informes de cobertura de código](#).

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Criterios de éxito/Cobertura de sucursales
- ***Pestaña de salidas/Informes/Configuración manual de informes/nombre-de-informe-1 /Criterios de éxito/Cobertura de sucursales***

Vulnerabilities

(nombre /Outputs/AutoDiscoverReports/SuccessCriteria/ Vulnerabilities de la acción)

Or (Disyunción)

(nombre de acción/Outputs/Reports/: nombre-informe 1) /SuccessCriteria/ Vulnerabilities

(Opcional)

Especifique el número máximo y la gravedad de las vulnerabilidades permitidas en el informe de la SCA para que el informe asociado CodeCatalyst se marque como aprobado. Para especificar las vulnerabilidades, debe especificar:

- La gravedad mínima de las vulnerabilidades que desea incluir en el recuento. Los valores válidos, del más grave al menos grave, son:CRITICAL,HIGH,MEDIUM,LOW,INFORMATIONAL.

Por ejemplo, si lo eligeHIGH, se HIGH contabilizarán CRITICAL las vulnerabilidades.

- El número máximo de vulnerabilidades de la gravedad especificada que desea permitir. Si se supera este número, el CodeCatalyst informe se marca como fallido. Los valores válidos son números enteros.

Los criterios de vulnerabilidad se aplican únicamente a los informes de la SCA. Para obtener más información sobre los informes de la SCA, consulte. [Informes de análisis de composición de software](#)

Para especificar la gravedad mínima, utilice la Severity propiedad. Para especificar el número máximo de vulnerabilidades, utilice la Number propiedad.

Para obtener más información sobre los informes de la SCA, consulte [Tipos de informes de prueba](#).

Interfaz de usuario correspondiente:

- Pestaña de salidas/Informes/Detección automática de informes/Criterios de éxito/Vulnerabilidades
- ***Pestaña de salidas/Informes/Configuración manual de informes/nombre-de-informe-1 /Criterios de éxito/ Vulnerabilidades***

Reports

(nombre de /Outputs/ Reports la acción)

(Opcional)

Una sección que especifica la configuración de los informes de pruebas.

Interfaz de usuario correspondiente: pestaña de salidas/informes

nombre-informe-1

(nombre de acción nombre-informe-1) /Outputs/Reports/

(Obligatorio si está incluido) [Reports](#)

El nombre que desea asignar al CodeCatalyst informe que se generará a partir de los informes sin procesar.

Interfaz de usuario correspondiente: pestaña de salidas/informes/configuración manual/nombre del informe

Format

*(nombre /Outputs/Reports/ de **Format** acción, nombre del informe, 1/)*

(Obligatorio si está incluido) [Reports](#)

Especifique el formato de archivo que va a utilizar para sus informes. Los valores posibles son los siguientes.

- Para los informes de prueba:

- Para Cucumber JSON, especifique Cucumber (editor visual) o CUCUMBERJSON (editor YAML).
- Para JUnit XML, especifique JUnit (editor visual) o JUNITXML (editor YAML).
- Para NUnit XML, especifique NUnit (editor visual) o **NUNITXML** (editor YAML).
- Para NUnit 3 XML, especifique NUnit3 (editor visual) o (editor YAML). NUNIT3XML
- Para Visual Studio TRX, especifique Visual Studio TRX (editor visual) o (editor YAML). VISUALSTUDIOTRX
- Para TestNG XML, especifique TestNG (editor visual) o TESTNGXML (editor YAML).
- Para los informes de cobertura de código:
 - Para Clover XML, especifique Clover (editor visual) o CLOVERXML (editor YAML).
 - Para Cobertura XML, especifique Cobertura (editor visual) o COBERTURAXML (editor YAML).
 - Para JaCoCo XML, especifique JaCoCo(editor visual) o JACOCOXML (editor YAML).
 - Para el SimpleCov JSON generado por [simplecov](#), no por [simplecov-json](#), especifique [Simplecov](#) (editor visual) o (editor YAML). SIMPLECOV
- Para los informes de análisis de composición de software (SCA):
 - Para SARIF, especifique SARIF (editor visual) o SARIFSCA (editor YAML).

Interfaz de usuario correspondiente: pestaña de salidas/Informes/ Configuración manual de informes/Añadir informe/ nombre-de-informe-1/ Tipo de informe y formato de informe

Configuration

(nombre de acción/) **Configuration**

(Obligatorio) Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

Steps

(nombre de *la acción /Configuration/ Steps*)

(Obligatorio)

Especifica tu código de GitHub acción tal como aparece en la página de detalles de la acción en [GitHub Marketplace](#). Agrega el código siguiendo estas pautas:

1. Pegue el código de la `steps` : sección de la GitHub acción en la `Steps` : sección del CodeCatalyst flujo de trabajo. El código comienza con un guión (-) y tiene un aspecto similar al siguiente.

GitHub código para pegar:

```
- name: Lint Code Base
  uses: github/super-linter@v4
  env:
    VALIDATE_ALL_CODEBASE: false
    DEFAULT_BRANCH: master
    GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

2. Revise el código que acaba de pegar y modifíquelo según sea necesario para que cumpla con los CodeCatalyst estándares. Por ejemplo, con el bloque de código anterior, puede quitar el código en ***cursiva roja*** y añadir el código en **negrita**.

CodeCatalyst flujo de trabajo yaml:

```
Steps:
  - name: Lint Code Base
    uses: github/super-linter@v4
    env:
      VALIDATE_ALL_CODEBASE: false
      DEFAULT_BRANCH: mastermain
      GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

3. Para el código adicional que se incluye en la GitHub acción pero que no existe dentro de la `steps` : sección, agrégalo al CodeCatalyst flujo de trabajo con un código CodeCatalyst equivalente. Puedes revisarla [Referencia de definición de flujo de trabajo](#) para obtener información sobre cómo puedes transferir tu GitHub código a CodeCatalyst. Los pasos de migración detallados están fuera del alcance de esta guía.

A continuación, se muestra un ejemplo de cómo especificar las rutas de los archivos en una acción de GitHub Actions:

```
Steps:
  - name: Lint Code Base
    uses: github/super-linter@v4
    ...
  - run: cd /sources/WorkflowSource/MyFolder/ && cat file.txt
```

```
- run: cd /artifacts/MyGitHubAction/MyArtifact/MyFolder/ && cat file2.txt
```

Para obtener más información sobre cómo especificar las rutas de los archivos, consulte [Hacer referencia a archivos de un repositorio fuente](#) y [Hacer referencia a los archivos de un artefacto](#).

Interfaz de usuario correspondiente: pestaña de GitHub configuración/acciones (YAML)

Referencia de acción «Renderizar la definición de tareas de Amazon ECS»

La siguiente es la referencia de YAML de definición de acciones para la acción de definición de tareas Render Amazon ECS. Para obtener información sobre cómo utilizar esta acción, consulte [Añadir la acción «Renderizar la definición de tareas de Amazon ECS»](#).

Note

La mayoría de las propiedades de YAML que aparecen a continuación tienen los elementos de interfaz de usuario correspondientes en el editor visual. Para buscar un elemento de la interfaz de usuario, usa Ctrl+F. El elemento aparecerá en la lista con su propiedad YAML asociada.

```
# The workflow definition starts here.
# See Propiedades de primer nivel for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSRenderTaskDefinition\_nn:
  Identifier: aws/ecs-render-task-definition@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
  Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
```

Artifacts:

- *task-definition-artifact*

Variables:

- Name: *variable-name-1*
Value: *variable-value-1*
- Name: *variable-name-2*
Value: *variable-value-2*

Configuration

task-definition: *task-definition-path*

container-definition-name: *container-definition-name*

image: *docker-image-name*

environment-variables:

- *variable-name-1=variable-value-1*
- *variable-name-2=variable-value-2*

Outputs:Artifacts:

- Name: *TaskDefArtifact*
- Files: "task-definition*"

Variables:

- *task-definition*

ECSRenderTaskDefinition

(Obligatorio)

Especifique el nombre de la acción. Todos los nombres de las acciones deben ser únicos en el flujo de trabajo. Los nombres de las acciones se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No se pueden usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las acciones.

Predeterminado: `ECSRenderTaskDefinition_nn`.

Interfaz de usuario correspondiente: pestaña de configuración/nombre de la acción

Identifier

(*ECSRenderTaskDefinition*/Identifier)

(Obligatorio)

Identifica la acción. No cambie esta propiedad a menos que desee cambiar la versión. Para obtener más información, consulte [Trabajar con versiones de acción](#).

Predeterminado: `aws/ecs-render-task-definition@v1`.

Interfaz de usuario correspondiente: diagrama de flujo de trabajo/ `ECSRenderTaskDefinition _nn/ aws/ @v1` label `ecs-render-task-definition`

DependsOn

(*ECSRenderTaskDefinition*/DependsOn)

(Opcional)

Especifique una acción o un grupo de acciones que debe ejecutarse correctamente para que esta acción se ejecute.

Para obtener más información sobre la funcionalidad «depende de», consulte [Configurar acciones para que dependan de otras acciones](#)

Interfaz de usuario correspondiente: pestaña de entradas/Depende de: opcional

Compute

(*ECSRenderTaskDefinition*/Compute)

(Opcional)

El motor informático utilizado para ejecutar las acciones del flujo de trabajo. Puede especificar el procesamiento en el nivel del flujo de trabajo o en el nivel de acción, pero no en ambos. Cuando se especifica en el nivel del flujo de trabajo, la configuración de procesamiento se aplica a todas las acciones definidas en el flujo de trabajo. En el nivel del flujo de trabajo, también puedes ejecutar varias acciones en la misma instancia. Para obtener más información, consulte [Compartir el cómputo entre las acciones](#).

Interfaz de usuario correspondiente: ninguna

Type

(*ECSRenderTaskDefinition*/Compute/Type)

(Obligatorio si [Compute](#) está incluido)

El tipo de motor de cómputo. Puede usar uno de los siguientes valores:

- EC2 (editor visual) o EC2 (editor YAML)

Optimizado para ofrecer flexibilidad durante las ejecuciones de acción.

- Lambda (editor visual) o Lambda (editor YAML)

Velocidades de inicio de acciones optimizadas.

Para obtener más información sobre los tipos de computación, consulte [Acerca de los tipos de procesamiento](#).

Interfaz de usuario correspondiente: pestaña de configuración/tipo de cálculo

Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Opcional)

Especifique la máquina o la flota que ejecutará el flujo de trabajo o las acciones del flujo de trabajo. Con las flotas bajo demanda, cuando se inicia una acción, el flujo de trabajo aprovisiona los recursos que necesita y las máquinas se destruyen cuando finaliza la acción. Ejemplos de flotas bajo demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obtener más información sobre las flotas bajo demanda, consulte. [Propiedades de la flota bajo demanda](#)

Con las flotas aprovisionadas, puede configurar un conjunto de máquinas dedicadas para ejecutar las acciones de su flujo de trabajo. Estas máquinas permanecen inactivas y listas para procesar las acciones de forma inmediata. Para obtener más información sobre las flotas aprovisionadas, consulte. [Propiedades de la flota aprovisionada](#)

Si Fleet se omite, el valor predeterminado es. `Linux.x86-64.Large`

Interfaz de usuario correspondiente: pestaña de configuración/flota de cómputo

Timeout

(*ECSRenderTaskDefinition*/Timeout)

(Opcional)

Especifica la cantidad de tiempo en minutos (editor YAML) o en horas y minutos (editor visual) que la acción puede ejecutarse antes de que CodeCatalyst finalice la acción. El mínimo es de 5 minutos y el máximo se describe en [Cuotas para flujos de trabajo en CodeCatalyst](#). El tiempo de espera predeterminado es el mismo que el tiempo de espera máximo.

Interfaz de usuario correspondiente: pestaña de configuración/tiempo de espera: opcional

Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Opcional)

La Inputs sección define los datos que se *ECSRenderTaskDefinition* necesitan durante la ejecución de un flujo de trabajo.

Note

Solo se permite una entrada (ya sea una fuente o un artefacto) por cada acción de definición de tarea de Render Amazon ECS. Las variables no se tienen en cuenta para este total.

Interfaz de usuario correspondiente: pestaña de entradas

Sources

(*ECSRenderTaskDefinition*/Inputs/Sources)

(Necesario si el archivo de definición de tareas está almacenado en un repositorio de origen)

Si el archivo de definición de tareas está almacenado en un repositorio de origen, especifique la etiqueta de ese repositorio de origen. Actualmente, la única etiqueta admitida es `WorkflowSource`.

Si el archivo de definición de tareas no está contenido en un repositorio de origen, debe residir en un artefacto generado por otra acción.

Para obtener más información sobre orígenes, consulte [Trabajar con fuentes](#).

Interfaz de usuario correspondiente: pestaña Entradas/Fuentes (opcional)

Artifacts - input

(*ECSRenderTaskDefinition*/Inputs/Artifacts)

(Obligatorio si el archivo de definición de tareas está almacenado en un [artefacto de salida](#) de una acción anterior)

Si el archivo de definición de tareas que desea implementar está contenido en un artefacto generado por una acción anterior, especifique ese artefacto aquí. Si el archivo de definición de tareas no está contenido en un artefacto, debe residir en el repositorio de origen.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de configuración/Artefactos: opcional

Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Obligatorio)

Especifique una secuencia de pares de nombre/valor que definan las variables de entrada que desea poner a disposición de la acción. Los nombres de las variables se limitan a caracteres alfanuméricos (a-z, A-Z, 0-9), guiones (-) y guiones bajos (_). No se permiten espacios. No puede usar comillas para habilitar los caracteres especiales y los espacios en los nombres de las variables.

Para obtener más información sobre las variables, incluidos ejemplos, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de entradas/variables: opcional

Configuration

(*ECSRenderTaskDefinition*/Configuration)

(Obligatorio)

Una sección en la que puede definir las propiedades de configuración de la acción.

Interfaz de usuario correspondiente: pestaña de configuración

task-definition

(*ECSRenderTaskDefinition*/Configuration/task-definition)

(Obligatorio)

Especifique la ruta a un archivo de definición de tareas existente. Si el archivo reside en el repositorio de origen, la ruta es relativa a la carpeta raíz del repositorio de origen. Si el archivo reside en un

artefacto de una acción de flujo de trabajo anterior, la ruta es relativa a la carpeta raíz del artefacto. Para obtener más información sobre los archivos de definición de tareas, consulte [Definiciones de tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/definición de tareas

container-definition-name

(*ECSRenderTaskDefinition*/Configuration/container-definition-name)

(Obligatorio)

Especifique el nombre del contenedor en el que se ejecutará la imagen de Docker. Puedes encontrar este nombre en el name campocontainerDefinitions, del archivo de definición de tareas. Para obtener más información, consulte [Nombre](#) en la guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/nombre del contenedor

image

(*ECSRenderTaskDefinition*/Configuration/image)

(Obligatorio)

Especifique el nombre de la imagen de Docker que desea que la acción de definición de tareas Render Amazon ECS añada al archivo de definición de tareas. La acción añade este nombre al image campocontainerDefinitions, del archivo de definición de tareas. Si ya existe un valor en el image campo, la acción lo sobrescribe. Puede incluir variables en el nombre de la imagen.

Ejemplos:

Si lo especificasMyDockerImage:\${WorkflowSource.CommitId}, la acción se agrega MyDockerImage: *commit-id* al archivo de definición de tareas, donde *commit-id es un ID* de confirmación generado en tiempo de ejecución por el flujo de trabajo.

Si lo especificasmy-ecr-repo/image-repo:\$(date +%m-%d-%y-%H-%m-%s), la acción añade my-ecr-repo/image-repo: date +%m-%d-%Y-%H-%m-%s al archivo de definición de tareas, donde se encuentra el URI de un Amazon Elastic Container Registry (ECR) y la fecha +%M-%D-%D-%Y-%H-%M-%s my-ecr-repos una marca de tiempo en el formato generado en tiempo de ejecución por el flujo de trabajo. month-day-year-hour-minute-second

Para obtener más información sobre el `image` campo, consulte la [imagen](#) en la guía para desarrolladores de Amazon Elastic Container Service. Para obtener más información sobre las variables, consulte [Trabajar con variables](#).

Interfaz de usuario correspondiente: pestaña de configuración/nombre de imagen

`environment-variables`

(*ECSRenderTaskDefinition*/Configuration/environment-variables)

(Obligatorio)

Especifique las variables de entorno que desee que la acción de definición de tareas Render Amazon ECS añada al archivo de definición de tareas. La acción añade las variables al `environment` campo `containerDefinitions`, del archivo de definición de tareas. Si las variables ya existen en el archivo, la acción sobrescribe los valores de las variables existentes y añade las variables nuevas. Para obtener más información sobre las variables de entorno de Amazon ECS, consulte [Especificación de variables de entorno](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Interfaz de usuario correspondiente: pestaña de configuración/variables de entorno (opcional)

Outputs

(*ECSRenderTaskDefinition*/Outputs)

(Obligatorio)

Define los datos que genera la acción durante la ejecución de un flujo de trabajo.

Interfaz de usuario correspondiente: pestaña de salidas

Artifacts

(*ECSRenderTaskDefinition*/Outputs/Artifacts)

(Obligatorio)

Especifique los artefactos generados por la acción. Puedes hacer referencia a estos artefactos como entrada en otras acciones.

Para obtener más información sobre los artefactos, incluidos ejemplos, consulte [Trabajar con artefactos](#).

Interfaz de usuario correspondiente: pestaña de salidas/artefactos

Name

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Name)

(Obligatorio)

Especifique el nombre del artefacto que contendrá el archivo de definición de tareas actualizado. El valor predeterminado es `MyTaskDefinitionArtifact`. A continuación, debe especificar este artefacto como entrada en la acción Implementar en Amazon ECS. Para saber cómo añadir este artefacto como entrada a la acción Implementar en Amazon ECS, consulte [Ejemplo de flujo de trabajo](#).

Interfaz de usuario correspondiente: pestaña de salidas/Artifectos/nombre

Files

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Files)

(Obligatorio)

Especifique los archivos que desee incluir en el artefacto. Debe `task-definition-*` especificarlo para que se incluya el archivo de definición de tareas actualizado `task-definition-`, que comienza con.

Interfaz de usuario correspondiente: pestaña de salidas/artefactos/archivos

Variables

(*ECSRenderTaskDefinition*/Outputs/Variables)

(Obligatorio)

Especifique el nombre de una variable que se establecerá mediante la acción de renderizado. La acción de renderización establecerá el valor de esta variable en el nombre del archivo de definición de tareas actualizado (por ejemplo, `task-definition-random-string.json`). A continuación, debe especificar esta variable en la propiedad Definición de tareas (editor visual) o `task-definition` (editor yaml) de la acción Implementar en Amazon ECS. Para saber cómo añadir esta variable a la acción Implementar en Amazon ECS, consulte [Ejemplo de flujo de trabajo](#).

Predeterminado: `task-definition`


Interfaz de usuario correspondiente: pestaña de salidas/variables/campo de nombre

Cuotas para flujos de trabajo en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de los flujos de trabajo en Amazon CodeCatalyst.

Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Número máximo de flujos de trabajo por espacio	800
Tamaño máximo del archivo de definición del flujo de trabajo	256 KB
Número máximo de archivos de flujo de trabajo procesados en un solo evento de origen	50
Número máximo de archivos procesados en un evento de origen único	4.000
Número máximo de flotas activas por espacio	10
Número máximo de instancias informáticas activas por flota	20
Número máximo de artefactos de entrada por acción	10
Número máximo de artefactos de salida por acción	10
Tamaño total máximo de las variables de salida de una sola acción	120 KB
Longitud máxima de un valor de variable de salida	500 caracteres o más, según la acción que emita el valor.

 Note

Los valores pueden truncarse si superan el límite de la acción.

Número máximo de días para conservar los artefactos generados durante la ejecución de un flujo de trabajo	30
Número máximo de informes por acción	50
Número máximo de casos de prueba por informe de prueba	20 000
Número máximo de archivos por informe de cobertura de código	20 000
Número máximo de resultados de análisis de composición de software por informe	20 000
Número máximo de archivos por informe de análisis estático	20 000
Número máximo de ejecuciones simultáneas de flujo de trabajo por espacio	100
Número máximo de acciones por flujo de trabajo	50
Número máximo de acciones que se ejecutan simultáneamente por flujo de trabajo	50
Número máximo de acciones que se ejecutan simultáneamente por espacio	200

Cantidad máxima de tiempo que puede ejecutarse una acción	<p>Para las acciones de compilación y prueba, el tiempo de espera es de 8 horas.</p> <p>Para todas las demás acciones, el tiempo de espera es de 1 hora.</p>
Número máximo de entornos asociados a una Cuenta de AWS por espacio	5 000
Número máximo de secretos por acción	5
Número máximo de secretos por espacio	500.000

Problemas en CodeCatalyst

En CodeCatalyst, puedes monitorear las características, los errores y cualquier otro trabajo relacionado con tu proyecto. Cada pieza de trabajo se guarda en un registro distinto denominado edición. Puedes dividir un tema en objetivos más pequeños añadiéndole una lista de tareas. Cada asunto puede tener una descripción, un responsable, un estado y otras propiedades que puede buscar, agrupar y filtrar. Puede ver sus problemas con las vistas predeterminadas o puede crear las suyas propias con filtros, clasificaciones o agrupaciones personalizados. Para obtener más información sobre los conceptos relacionados con los problemas, consulte [Conceptos de problemas](#). Para obtener información sobre cómo crear su primer número, consulte [Crear un problema en CodeCatalyst](#).

Este es un posible flujo de trabajo para un equipo que utiliza problemas:

Jorge Souza es un desarrollador que trabaja en un proyecto. Él y sus compañeros del proyecto Li Juan, Mateo Jackson y Wang Xiulan colaboran para determinar qué trabajo hay que hacer. Todos los días, él y sus colegas desarrolladores celebran una reunión de sincronización dirigida por Wang Xiulan. Para abrir el tablero, acceden a una de las vistas de su equipo sobre el tablero. Al crear vistas, los usuarios y los equipos pueden guardar los filtros, las agrupaciones y la clasificación de las cuestiones para ver fácilmente las cuestiones que cumplen los criterios especificados. Su vista contiene los problemas agrupados por responsable y ordenados por prioridad para mostrar los problemas más importantes y el estado de los problemas para cada desarrollador. A medida que a Jorge se le asignan tareas para completar, planifica su trabajo creando una incidencia para cada tarea. Al crear las incidencias, Jorge puede elegir el estado, la prioridad y el esfuerzo de estimación del trabajo adecuados. En el caso de las cuestiones más importantes, Jorge añade tareas a la cuestión para dividir el trabajo en objetivos más pequeños. Jorge crea sus problemas con un estado de borrador, por ejemplo, pendiente, ya que no planea empezar con ellos de inmediato. Las cuestiones en estado de borrador aparecen en la vista de borradores, donde se deben planificar y priorizar. Una vez que Jorge esté listo para empezar el trabajo, mueve el tema correspondiente al tablero actualizando su estado a un estado de otra categoría (No iniciado, Empezado o Completado). A medida que se trabaja en cada tarea, el equipo puede filtrar por título, estado, responsable, etiqueta, prioridad y estimación para encontrar un problema específico o problemas similares que coincidan con el parámetro especificado. Mediante el tablero, Jorge y su equipo pueden ver el número de tareas completadas en cada asunto y hacer un seguimiento del day-to-day progreso arrastrando cada asunto de un estado a otro hasta completar la tarea. A medida que el proyecto avanza, las cuestiones finalizadas se acumulan en el estado Completado. Wang Xiulan

decide eliminarlos de la vista archivándolos con el botón de archivado rápido, de modo que los desarrolladores puedan centrarse en los problemas relacionados con el trabajo actual y futuro.

A la hora de planificar su trabajo, los desarrolladores que trabajan en el proyecto eligen Ordenar por y agrupar por para buscar las cuestiones que desean pasar del trabajo pendiente al tablero. Pueden optar por añadir temas al foro en función de las solicitudes de los clientes con mayor prioridad, por lo que agrupan el foro por una etiqueta de solicitud del cliente y las ordenan por prioridad. También pueden ordenarlos por estimación para asegurarse de que están asumiendo un volumen de trabajo que pueden realizar. El director del proyecto, Saanvi Sarkar, revisa y corrige periódicamente el trabajo pendiente para garantizar que la prioridad refleje con precisión la importancia de cada tema para el éxito del proyecto.

Temas

- [Conceptos de problemas](#)
- [Crear un problema en CodeCatalyst](#)
- [Edición y colaboración en temas de CodeCatalyst](#)
- [Problemas de búsqueda y visualización](#)
- [Problemas de exportación](#)
- [Configuración de los ajustes de problemas](#)
- [Cuotas para emisiones en CodeCatalyst](#)

Conceptos de problemas

Crear una edición es una forma rápida y eficaz de hacer un seguimiento del trabajo que se está realizando en un proyecto. Puedes usar los temas como ayuda para analizar el trabajo en las reuniones diarias de sincronización, priorizar el trabajo y mucho más.

Esta página incluye una lista de conceptos que te ayudarán a utilizar los problemas de forma eficaz. CodeCatalyst

Problemas activos

Las cuestiones activas son aquellas que no están en estado de borrador ni archivadas. En otras palabras, las cuestiones activas son aquellas que tienen un estado en cualquiera de las siguientes

categorías de estado: No iniciadas, Iniciadas y Finalizadas. Para obtener más información sobre los estados y las categorías de estado, consulte [Estado y categorías de estado](#).

Puede ver todos los problemas activos de su proyecto desde la vista de problemas activos predeterminada.

Problemas archivados

Un problema archivado es un problema que ya no es relevante para su proyecto. Por ejemplo, puedes [archivar un problema](#) si está completo y ya no necesitas verlo en la columna Listo o si se creó por error. Los problemas archivados se pueden desarchivar si es necesario.

Cesionario

El cesionario es la persona a la que se le asigna la emisión. Si la persona no aparece en la lista cuando la buscas, significa que no se ha agregado a tu proyecto. Para añadirlos, consulte [Invitar a un usuario a tu proyecto](#). Para permitir la participación de varias personas asignadas a una incidencia, consulte [Habilitar o deshabilitar varios cesionarios](#). Los problemas con varios responsables aparecerán en tu tablero con avatares de diferentes colores, cada uno de los cuales representa a uno de los responsables.

Campos personalizados

Los campos personalizados te permiten personalizar los distintos atributos de una incidencia según tus necesidades de seguimiento y mantenimiento de las incidencias de un proyecto. Por ejemplo, puedes añadir un campo para trazar una hoja de ruta, una fecha de entrega específica o un campo de solicitante.

Estimación

En el desarrollo ágil, la estimación se conoce como puntos de la historia. Utilice la estimación para incluir la cantidad de trabajo necesaria, además de la ambigüedad y la complejidad del problema. Utilice estimaciones más altas para los problemas con mayor riesgo, dificultad e incógnitas. Puede cambiar el tipo de estimación de su proyecto. Para obtener más información sobre los tipos de estimación y cómo configurarlos, consulte [Configurar la estimación del esfuerzo del problema](#).

Problema

Un problema es un registro que hace un seguimiento del trabajo relacionado con el proyecto. Puedes crear un problema para una función, una tarea, un error o cualquier otra obra relacionada con tu

proyecto. Si utilizas un desarrollo ágil, un problema también puede describir una historia épica o de usuario.

Etiqueta

La etiqueta se usa para agrupar, ordenar y filtrar problemas. Puede introducir un nombre de etiqueta nuevo o elegir una de las etiquetas de la lista completa. Esta lista incluye las etiquetas utilizadas recientemente en el proyecto. Una edición puede tener varias etiquetas y una etiqueta se puede quitar de una edición. Para personalizar las etiquetas, consulte [Etiquetas](#).

Priority (Prioridad)

La prioridad se refiere al nivel de importancia del tema. Hay cuatro opciones: baja, media, alta y sin prioridad.

Estado y categorías de estado

El estado es el estado actual del problema y se utiliza para comprobar rápidamente el progreso de un problema a lo largo de su ciclo de vida, desde su inicio hasta su finalización. Todos los problemas deben tener un estado y cada estado pertenece a una categoría de estado. Las categorías de estado se utilizan para ayudar a organizar los estados y completar las vistas de problemas predeterminadas.

Hay cinco estados predeterminados y cuatro categorías de estado. CodeCatalyst Puede crear otros estados, pero no puede crear otras categorías de estado. La siguiente lista contiene los estados predeterminados y sus categorías de estado entre paréntesis: pendiente (borrador), pendiente (no iniciado), En curso (iniciado), En revisión (iniciado) y Listo (completado).

Para obtener más información sobre cómo trabajar con estados, consulte [Estados](#)

Tareas

Se pueden añadir tareas a las cuestiones para desglosar y organizar aún más el trabajo de cada cuestión. Puedes añadir tareas a un asunto al crearlo o añadir tareas a un asunto existente. Al ver un problema, puedes reordenar, eliminar o marcar sus tareas como completadas.

Vistas

Los problemas del CodeCatalyst proyecto se muestran en las vistas. Las vistas pueden ser vistas de cuadrícula que muestran los problemas en formato de lista o vistas de tablero que muestran los

problemas en forma de mosaicos en columnas organizados por estado de los problemas. Hay cuatro vistas predeterminadas y puedes [crear tus propias vistas con agrupaciones, filtros y clasificaciones personalizados](#). La siguiente lista contiene detalles sobre las cuatro vistas predeterminadas.

- La vista de borradores es una vista de cuadrícula que muestra los problemas en los que no se está trabajando actualmente. En esta vista se muestra cualquier asunto creado con un estado en la categoría de estado Borrador. Los equipos pueden usar esta vista para ver qué problemas aún se están definiendo o están pendientes de ser asignados y resueltos.
- La vista de problemas activos es una vista general de todos los problemas en los que se está trabajando actualmente. En esta vista se mostrará cualquier problema con un estado en las categorías de estado No iniciado, Iniciado o Finalizado.
- La vista de todos los problemas es una vista de cuadrícula que muestra todos los problemas del proyecto, tanto los borradores como los problemas activos.
- La vista archivada muestra todos los problemas archivados.

Crear un problema en CodeCatalyst

Las funciones de IA generativa de Amazon CodeCatalyst se encuentran en versión preliminar y están sujetas a cambios. Solo están disponibles en la región EE.UU. Oeste (Oregón). El acceso a las funciones generativas de IA varía según el nivel. Para más información, consulte [Precios de](#) .

Los equipos de desarrollo crean problemas para facilitar el seguimiento y la gestión de su trabajo. Puede crear problemas dentro de un proyecto en función de sus necesidades. Por ejemplo, puedes crear una incidencia para hacer un seguimiento de la actualización de una variable de tu código. Puedes asignar problemas a otros usuarios del proyecto, usar etiquetas para ayudarte a realizar un seguimiento de tu trabajo y mucho más.

Sigue estas instrucciones para crear una edición en CodeCatalyst.

Para crear una incidencia

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el proyecto en el que desee crear una edición.
3. En la página de inicio del proyecto, selecciona Crear edición. Como alternativa, en el panel de navegación, selecciona Problemas.

4. Selecciona Crear problema.

Note

También puedes añadir las incidencias en línea cuando utilices una vista de cuadrícula.


5. Introduce un título para el número.
6. (Opcional) Introduce una descripción. Puedes usar Markdown para añadir formato.
7. (Opcional) Elige un estado, una prioridad o una estimación para el problema. Si la configuración de estimación del proyecto está configurada en Ocultar estimaciones, no habrá ningún campo de estimación.
8. (Opcional) Agrega tareas al problema. Las tareas se pueden usar para dividir el trabajo de un tema en objetivos más pequeños. Para añadir una tarea, selecciona + Añadir tareas. Luego, ingresa el nombre de la tarea en el campo de texto y presiona enter. Después de añadir las tareas, puedes marcarlas como completadas seleccionando la casilla de verificación o reordenándolas seleccionando y arrastrando la tarea desde el lado izquierdo de la casilla de verificación.
9. (Opcional) Añade una etiqueta existente o crea una nueva y agrégala seleccionando + Añadir etiqueta.
 - a. Para añadir una etiqueta existente, selecciónela de la lista. Puede introducir un término de búsqueda en el campo para buscar en el proyecto todas las etiquetas que contengan ese término.
 - b. Para crear una etiqueta nueva y añadirla, introduce el nombre de la etiqueta que deseas crear en el campo de búsqueda y pulsa enter.
10. (Opcional) Agregue un cesionario seleccionando + Agregar un cesionario. Puedes añadirte rápidamente como cesionario seleccionando + Añadirme.

Tip

Puedes asignar un problema a Amazon Q para que Amazon Q intente resolverlo. Para obtener más información, consulte [Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo](#).

Esta funcionalidad requiere que las funciones de IA generativa estén habilitadas para el espacio. Para obtener más información, consulte [Administrar las funciones de IA generativa](#).

11. (Opcional) Agregue un campo personalizado existente o cree uno nuevo. Los problemas pueden tener varios campos personalizados.
 - a. Para añadir un campo personalizado existente, elija el campo personalizado de la lista. Puede introducir un término de búsqueda en el campo para buscar en todos los campos personalizados que contengan ese término en el proyecto.
 - b. Para crear un campo personalizado nuevo y añadirlo, introduzca el nombre del campo personalizado que desee crear en el campo de búsqueda y pulse Entrar. A continuación, elija el tipo de campo personalizado que desee crear y establezca un valor.
12. Selecciona Crear incidencia. Aparece una notificación en la esquina inferior derecha: si el problema se creó correctamente, aparecerá un mensaje de confirmación que indica que el problema se creó correctamente. Si el problema no se ha creado correctamente, aparece un mensaje de error con el motivo del error. A continuación, puede seleccionar Reintentar para editar y volver a intentar crear el problema, o bien seleccionar Descartar para descartar el problema. Ambas opciones descartarán la notificación.

 Note

No puedes vincular una solicitud de cambios a una edición al crearla. Sin embargo, puedes [editarla](#) después de crearla para añadir enlaces a las solicitudes de cambios.

Prácticas recomendadas a la hora de crear problemas asignados a Amazon Q y trabajar con ellos

Al crear problemas, a veces algunos de ellos persisten. Las causas de esto pueden ser complejas y variables. A veces se debe a que no está claro quién debe trabajar en ello. Otras veces, el tema requiere investigación o experiencia con una parte determinada del código base y los mejores candidatos para el trabajo están ocupados con otros temas. A menudo hay otro trabajo urgente que hay que atender primero. Cualquiera o todas estas causas pueden provocar problemas que no se resuelvan. CodeCatalyst incluye la integración con un asistente de IA generativa llamado Amazon Q que puede analizar un problema en función de su título y su descripción. Si asignas el problema a Amazon Q, intentará crear un borrador de solución para que lo evalúes. Esto puede ayudarte a ti y a tu equipo a centrar y optimizar el trabajo en los problemas que requieren tu atención, mientras que Amazon Q trabaja en una solución para los problemas que no tienes recursos para abordar de forma inmediata.

 Note

Desarrollado por Amazon Bedrock: AWS implementa la [detección automática de abusos](#). Como la capacidad de desarrollo de funciones Asignar problemas a Amazon Q está integrada en Amazon Bedrock, los usuarios pueden aprovechar al máximo los controles implementados en Amazon Bedrock para garantizar la protección, la seguridad y el uso responsable de la inteligencia artificial (IA).

Amazon Q funciona mejor en temas simples y problemas directos. Para obtener los mejores resultados, utilice un lenguaje sencillo para explicar claramente lo que quiere que se haga. Las siguientes son algunas prácticas recomendadas que te ayudarán a crear problemas optimizados para que Amazon Q.

- Manténgalo sencillo. Amazon Q funciona mejor con cambios y correcciones de código simples que se pueden explicar en el título y la descripción del problema. No asigne a los problemas títulos vagos o descripciones demasiado floreadas o contradictorias.
- Sé específico. Cuanta más información puedas proporcionar sobre los cambios exactos necesarios para resolver el problema, más probabilidades habrá de que Amazon Q pueda crear una solución que resuelva el problema. Si es posible, incluya detalles específicos, como el nombre de las API que desee cambiar, los métodos que desee actualizar, las pruebas que necesiten cambios y cualquier otro detalle que se le ocurra.
- Asegúrate de incluir todos los detalles en el título y la descripción del problema antes de asignarlo a Amazon Q. No puedes cambiar el título o la descripción de un número después de asignarlo a Amazon Q, así que asegúrate de tener toda la información requerida en un número antes de asignarlo a Amazon Q.
- Asigne únicamente los problemas que requieran cambios de código a un único repositorio fuente. Amazon Q solo puede funcionar con el código de un único repositorio fuente en CodeCatalyst. No se admiten los repositorios enlazados. Asegúrese de que el problema solo requiera cambios en un repositorio de origen único antes de asignarlo a Amazon Q.
- Usa la opción predeterminada sugerida por Amazon Q para aprobar cada paso. De forma predeterminada, Amazon Q requerirá tu aprobación para cada paso que dé. Esto te permite interactuar con Amazon Q en los comentarios no solo sobre el tema, sino también sobre cualquier solicitud de cambios que cree. Esto proporciona una experiencia más interactiva con Amazon Q que le ayuda a ajustar su enfoque y refinar el código que crea para resolver el problema.

Note

Amazon Q no responde a los comentarios individuales de las ediciones o solicitudes de cambios, pero los revisará cuando se le pida que reconsidere su enfoque o cree una revisión.

- Revise siempre detenidamente el enfoque sugerido por Amazon Q. Una vez que apruebe su enfoque, Amazon Q empezará a trabajar en la generación de código basado en ese enfoque. Asegúrese de que el enfoque le parezca correcto e incluya todos los detalles que espera antes de decirle a Amazon Q que continúe.
- Asegúrate de permitir que Amazon Q solo funcione en flujos de trabajo si no tienes flujos de trabajo existentes que puedan implementarlos antes de revisarlos. Es posible que tu proyecto tenga flujos de trabajo configurados para iniciar ejecuciones en eventos de solicitudes de incorporación de cambios. Si es así, cualquier solicitud de extracción que Amazon Q cree y que incluya la creación o actualización del flujo de trabajo YAML podría iniciar una ejecución de los flujos de trabajo incluidos en la solicitud de extracción. Como práctica recomendada, no permitas que Amazon Q trabaje en archivos de flujo de trabajo a menos que estés seguro de que no hay flujos de trabajo en tu proyecto que los ejecuten automáticamente antes de revisar y aprobar la solicitud de extracción que crea.

Para obtener más información, consulte [Gestión de Tutorial: Uso de funciones de IA CodeCatalyst generativa para acelerar el trabajo de desarrollo las funciones de IA generativa](#).

Edición y colaboración en temas de CodeCatalyst

Contenido

- [Edición de una edición](#)
- [Uso de archivos adjuntos](#)
- [Administrar las tareas relacionadas con los problemas](#)
- [Marcar un problema como bloqueado o desbloqueado](#)
- [Añadir, editar o eliminar comentarios](#)
 - [Usar menciones en un comentario](#)
- [¿Está progresando con un problema](#)
 - [Entre el trabajo pendiente y el tablero](#)

- [Progresar un tema a lo largo de las etapas del ciclo de vida del foro](#)
- [Mover problemas entre grupos](#)
- [Archivar una incidencia](#)

Edición de una edición

Sigue estos pasos para editar el título, la descripción, el estado, el responsable, la prioridad, la estimación o las etiquetas de un problema.

Para editar un número

1. Elige el problema que deseas editar para ver los detalles del problema. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
2. Para editar el título del número, selecciónelo, introduzca uno nuevo y pulse Entrar.
3. Para editar la descripción, selecciónela, introduzca una nueva descripción y pulse Entrar. Puedes usar Markdown para añadir formato.
4. En Tareas, puedes ver y gestionar las tareas del problema. Para obtener más información, consulte [Administrar las tareas relacionadas con los problemas](#).
5. Para editar el estado, la estimación o la prioridad, selecciona una opción de los menús desplegados correspondientes.
6. En Etiquetas, puedes añadir una etiqueta existente, crear una nueva etiqueta o eliminar una etiqueta.
 - a. Para añadir una etiqueta existente, selecciona + Añadir etiqueta y elige la etiqueta de la lista. Puede introducir un término de búsqueda en el campo para buscar en el proyecto todas las etiquetas que contengan ese término.
 - b. Para crear una etiqueta nueva y añadirla, seleccione + Añadir etiqueta, introduzca el nombre de la etiqueta que desee crear en el campo de búsqueda y pulse Entrar.
 - c. Para eliminar una etiqueta, selecciona el icono X situado junto a la etiqueta que quieres eliminar. Si eliminas una etiqueta de todas las ediciones, la etiqueta aparecerá en la sección Etiquetas no utilizadas de la sección Etiquetas de la configuración de la edición. Las etiquetas no utilizadas aparecen al final de la lista de etiquetas cuando se utilizan filtros o se añaden etiquetas a una edición. Puedes encontrar un resumen de todas las etiquetas (usadas y no utilizadas) y de las ediciones que las contienen en la configuración de ediciones.

7. Para asignar una incidencia, selecciona + Añadir un responsable en la sección Responsable y, a continuación, busca y selecciona al responsable de la lista. Puedes elegir + Añadirme para añadirte rápidamente como cesionario.
8. En Adjuntos, puedes añadir, descargar o eliminar archivos adjuntos. Para obtener más información, consulte [Uso de archivos adjuntos](#).
9. Para vincular una solicitud de extracción, selecciona Enlazar una solicitud de extracción y, a continuación, elige una solicitud de extracción de la lista o introduce su URL o ID. Para desvincular una solicitud de extracción, selecciona el icono de desvinculación.

Tip

Después de añadir un enlace a una solicitud de atracción de una publicación, puedes acceder rápidamente a ella eligiendo su ID en la lista de solicitudes de extracción vinculadas. Puedes usar la URL de una solicitud de cambios para vincular las solicitudes de cambios que estén en proyectos distintos del foro de emisión, pero solo los usuarios que sean miembros de ese proyecto podrán ver esa solicitud de cambios o navegar hasta ella.

10. (Opcional) Agrega y establece un campo personalizado existente, crea un campo personalizado nuevo o elimina un campo personalizado. Los problemas pueden tener varios campos personalizados.
 - a. Para añadir un campo personalizado existente, elija el campo personalizado de la lista. Puede introducir un término de búsqueda en el campo para buscar en todos los campos personalizados que contengan ese término en el proyecto.
 - b. Para crear un campo personalizado nuevo y añadirlo, introduzca el nombre del campo personalizado que desee crear en el campo de búsqueda y pulse Entrar. A continuación, elija el tipo de campo personalizado que desee crear y establezca un valor.
 - c. Para eliminar un campo personalizado, selecciona el icono X situado junto al campo personalizado que deseas eliminar. Si eliminas un campo personalizado de todos los números, el campo personalizado se eliminará y ya no lo verás al filtrar.

Uso de archivos adjuntos

Puede añadir archivos adjuntos a los números CodeCatalyst para facilitar el acceso a los archivos relacionados. Utilice el siguiente procedimiento para gestionar los archivos adjuntos de un problema.

El tamaño de los archivos adjuntos que se añaden a las ediciones se tiene en cuenta a la hora de calcular las cuotas de almacenamiento de tu espacio. Para obtener información sobre cómo ver y administrar los archivos adjuntos de su proyecto, consulte [Visualización y administración de los archivos adjuntos](#).

⚠ Important

Amazon no escanea ni analiza los archivos adjuntos a los números CodeCatalyst. Cualquier usuario puede añadir un archivo adjunto a una publicación que pueda contener código o contenido malintencionado. Asegúrese de que los usuarios conozcan las mejores prácticas en lo que respecta a la administración de los archivos adjuntos y a la protección contra códigos, contenidos o virus maliciosos.

Para añadir, descargar o eliminar archivos adjuntos

1. Elija el tema para el que desee administrar los archivos adjuntos. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
2. Para añadir un archivo adjunto, selecciona Cargar archivo. Navega hasta el archivo en el explorador de archivos de tu sistema operativo y selecciónalo. Selecciona Abrir para añadirlo como archivo adjunto. Para obtener información sobre las cuotas, como el tamaño máximo de los archivos adjuntos, consulte [Cuotas para emisiones en CodeCatalyst](#).

Tenga en cuenta las siguientes restricciones en cuanto a los nombres de los archivos adjuntos y a los tipos de contenido:

- Los siguientes caracteres no están permitidos en los nombres de los archivos:
 - Caracteres de control: `0x00-0x1f` y `0x80-0x9f`
 - Caracteres reservados: `/?`, `<>`, `\`, `:`, `*`, `|`, y `"`
 - Nombres de archivos reservados de Unix: `y . . .`
 - Periodos y espacios finales
 - Nombres de archivos reservados de Windows: `CON`, `PRN`, `AUX`, `NUL`, `COM1`, `COM2`, `COM3`, `COM4`, `COM5`, `COM6`, `COM7`, `COM8`, `COM9`, `LPT1`, `LPT2`, `LPT3`, `LPT4`, `LPT5`, `LPT6`, `LPT7`, `LPT8`, and `LPT9`
- El tipo de contenido del archivo adjunto debe seguir el siguiente patrón de tipos de soporte:

```
media-type = type "/" [tree "."] subtype ["+" suffix]* [";" parameter];
```

Por ejemplo, `text/html; charset=UTF-8`.

3. Para descargar un archivo adjunto, seleccione el menú de puntos suspensivos situado junto al archivo adjunto que desee descargar y seleccione Descargar.
4. Para copiar la URL de un archivo adjunto, selecciona el menú de puntos suspensivos situado junto al archivo adjunto del que quieres copiar la URL y selecciona Copiar URL.
5. Para eliminar un archivo adjunto, selecciona el menú de puntos suspensivos situado junto al archivo adjunto que quieres eliminar y selecciona Eliminar.

Administrar las tareas relacionadas con los problemas

Se pueden añadir tareas a las cuestiones para desglosar, organizar y realizar un seguimiento del trabajo de cada cuestión.

Para gestionar las tareas relacionadas con un tema

1. Elige el asunto para el que quieres gestionar las tareas. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
2. En Tareas, puede ver y gestionar las tareas relacionadas con el problema.
 1. Para añadir una tarea, introduce el nombre de la tarea en el campo de texto y pulsa Entrar.
 2. Para marcar una tarea como completada, selecciona la casilla de verificación de la tarea.
 3. Para ver o actualizar los detalles de una tarea, selecciónela de la lista.
 4. Para reordenar las tareas, selecciona y arrastra la tarea desde el lado izquierdo de la casilla de verificación.
 5. Para eliminar una tarea, selecciona el menú de puntos suspensivos de la tarea y selecciona Eliminar.

Marcar un problema como bloqueado o desbloqueado

Si hay algo que te impide solucionar un problema, puede que desees marcarlo como bloqueado. Por ejemplo, el problema podría estar bloqueado si se debe a un cambio en otra parte de la base de código que aún no se ha fusionado.

Cuando marcas una publicación como bloqueada, CodeCatalyst añade una etiqueta roja de Bloqueado a la publicación, lo que la hace muy visible en tu cartera de trabajo, en tu archivo o en tu tablero.

Puedes desbloquear el problema cuando se resuelvan las circunstancias externas.

Para marcar un problema como bloqueado

1. Abre el problema que quieres marcar como bloqueado. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
2. Selecciona Acciones y, a continuación, selecciona Marcar como bloqueado.

Para desbloquear un problema

1. Abre el problema que quieres desbloquear. Si necesitas ayuda para encontrar el problema, consulta [Problemas de búsqueda y visualización](#).
2. Selecciona Acciones y, a continuación, selecciona Marcar como desbloqueado.

Añadir, editar o eliminar comentarios

Puedes dejar un comentario sobre un asunto. En los comentarios puedes etiquetar a otros miembros del espacio, otros proyectos del espacio, temas relacionados y código.

Para añadir un comentario a un número

1. Vaya a su proyecto.
2. En la barra de navegación, selecciona Problemas.
3. Elige el tema en el que quieres añadir el comentario. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
4. Introduzca el comentario en el campo Comentarios. Puedes usar Markdown para añadir formato.
5. Seleccione Enviar.

Para editar un comentario

Puedes editar los comentarios que hagas sobre los temas. Solo puedes editar los comentarios de tu autoría.

1. Vaya a su proyecto.
2. En la barra de navegación, selecciona Problemas.
3. Elige el tema en el que quieres editar un comentario. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
4. Para editar un comentario, busca el comentario que deseas editar.

 Tip

Puedes ordenar primero los comentarios por los más antiguos o los más recientes. Los comentarios se cargan de 10 en 10.

5. Seleccione el icono de puntos suspensivos y, a continuación, seleccione Editar.
6. Edita el comentario. Puedes usar Markdown para añadir formato.
7. Seleccione Guardar. El comentario ya está actualizado.

Para eliminar un comentario

Puedes eliminar los comentarios que hagas sobre temas. Solo puedes eliminar los comentarios de tu autoría. Al eliminar un comentario, aparecerá tu nombre de usuario, pero con las palabras Este comentario se ha eliminado en lugar del texto original del comentario.

1. Vaya a su proyecto.
2. En la barra de navegación, selecciona Problemas.
3. Elige el tema del que quieres eliminar un comentario. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
4. Selecciona el icono de puntos suspensivos, selecciona Eliminar y, a continuación, selecciona Confirmar.

Usar menciones en un comentario

Puedes mencionar a los miembros del espacio, otros proyectos del espacio, temas relacionados y código en los comentarios. Al hacerlo, se crea un enlace rápido al usuario o recurso que mencionas.

A @mention en un comentario

1. Vaya a su proyecto.

2. En la barra de navegación, selecciona Problemas.
3. Elige el problema que deseas editar para ver los detalles del problema. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
4. Selecciona el cuadro de texto Añadir un comentario.
5. Escribe `@user_name` para mencionar a otro usuario.
6. Escribe `@project_name` para mencionar un proyecto.
7. Escribe `@issue_name` o `@issue_number` mencione otro problema.
8. Escribe `@file_name` para mencionar archivos o códigos específicos en un repositorio de código fuente.

Note

A medida que escribas, aparecerá una lista de los 5 elementos principales (usuarios, repositorios de fuentes, proyectos, etc.) que contenga los términos que coincidan con los tuyos@mention.

9. Elige el elemento que deseas mencionar. La ruta que muestra la ubicación del elemento aparecerá en el cuadro de texto del comentario.
10. Termina tu comentario y selecciona Enviar.

¿Está progresando con un problema

Cada problema tiene un ciclo de vida. En CodeCatalyst, los problemas suelen empezar como un borrador pendiente. Cuando se va a empezar a trabajar en esa edición, se pasa a otra categoría de estado y pasa por varios estados hasta que se completa y, a continuación, se archiva. Puede mover o hacer avanzar una incidencia a lo largo de su ciclo de vida de las siguientes maneras:

- Puedes mover un asunto entre la lista de temas pendientes y el foro.
- Puedes hacer que el tema en curso pase por varias etapas de finalización.
- Puedes archivar una incidencia que esté completa.

Entre el trabajo pendiente y el tablero

Una vez que empieces a trabajar en el tema, puedes pasar un tema del trabajo pendiente al foro. Si el trabajo se pospone, también puedes volver a incluir un tema en la lista de trabajos pendientes.

Para trasladar una cuestión entre el trabajo pendiente y el foro

1. Vaya a su proyecto.
2. En el panel de navegación, selecciona Problemas. La vista predeterminada es el tablero.
3. Para trasladar una cuestión del foro a la lista de temas pendientes:
 - a. Elige el asunto que quieres mover. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
 - b. Selecciona Backlog en el menú desplegable de estado.
4. Para mover un asunto de la lista de temas pendientes al foro:
 - a. Para acceder a la lista de pendientes, selecciona Board y selecciona Backlog.
 - b. Elige el tema que quieres trasladar. Si necesitas ayuda para encontrar tu problema, consulta [Problemas de búsqueda y visualización](#).
 - c. Selecciona Añadir al tablero o elige un estado que no sea el de trabajo pendiente.

Progresar un tema a lo largo de las etapas del ciclo de vida del foro

Puedes mover un tema de un tablero a través de diferentes estados hasta que se complete.

Para mover una cuestión dentro del tablero

1. En el panel de navegación, selecciona Problemas. La vista predeterminada es el tablero.
2. Realice una de las acciones siguientes:
 - Arrastra y suelta un tema para colocarlo en otro estado.
 - Selecciona un problema y, a continuación, selecciona un estado en el menú desplegable Estado.
 - Elige un asunto y, a continuación, selecciona Mover al ***siguiente estado***.

Para obtener información sobre cómo archivar una edición, consulte. [Archivar una incidencia](#)

Mover problemas entre grupos

Puedes [agrupar los problemas](#) en las vistas Todos los temas y del Consejo según varios parámetros. Si las cuestiones están agrupadas, puedes moverlas de un grupo a otro. Al mover una incidencia

de un grupo a otro, se editará automáticamente el campo en el que están agrupadas las incidencias para que coincida con el grupo objetivo.

A modo de ejemplo, supongamos que hay una empresa CodeCatalyst que utiliza archivos y que tiene temas asignados a dos personas, Wang Xiulan y Saanvi Sarkar. El foro está agrupado por grupos y hay dos grupos `Assignee`, uno para cada responsable. Si se traslada una emisión del grupo Wang Xiulan al grupo Saanvi Sarkar, el cesionario de la emisión pasará a ser Saanvi Sarkar.

Archivar una incidencia

Note

Los problemas no se eliminan de un proyecto, sino que se archivan. Para eliminar los problemas, debe eliminar el proyecto.

Puede archivar un problema cuando ya no lo necesite en su proyecto. Al archivar un problema, lo CodeCatalyst elimina de todas las vistas que filtran los problemas archivados. Los problemas archivados se pueden ver en la vista predeterminada de problemas archivados, donde se pueden desarchivar si es necesario.

Puede archivar un problema si:

- Has completado el problema y ya no lo necesitas en la columna Listo.
- No tienes planes de trabajar en ello.
- Lo creaste por error.
- Has alcanzado el número máximo de incidencias activas.

Para archivar un problema

1. Abre el asunto que deseas archivar. Si necesitas ayuda para encontrar el problema, consulta [Problemas de búsqueda y visualización](#).
2. Selecciona Acciones y, a continuación, selecciona Mover a archivar.
3. (Opcional) Para archivar rápidamente varios números con el estado Completado, selecciona los puntos suspensivos verticales situados en la parte superior de cualquier estado Completado de la pizarra y selecciona Archivar números.

Para desarchivar un problema

1. Abre el problema que deseas desarchivar. Para ver una lista de incidencias archivadas, abre la vista de incidencias archivadas en el menú desplegable del selector de incidencias. Si necesitas ayuda para encontrar tu problema, consulta. [Problemas de búsqueda y visualización](#)
2. Selecciona Desarchivar.

Problemas de búsqueda y visualización

En las siguientes secciones se describe cómo buscar y ver de forma eficaz los problemas de un CodeCatalyst proyecto.

Los problemas CodeCatalyst del proyecto se muestran en las vistas. Las vistas pueden ser vistas de cuadrícula que muestran los problemas en formato de lista o vistas de tablero que muestran los problemas en forma de mosaicos en columnas organizados por estado de los problemas. Hay cuatro vistas predeterminadas y puedes [crear tus propias vistas con agrupaciones, filtros y clasificaciones personalizados](#). La siguiente lista contiene detalles sobre las cuatro vistas predeterminadas.

- La vista de borradores es una vista de cuadrícula que muestra los problemas en los que no se está trabajando actualmente. En esta vista se muestra cualquier asunto creado con un estado en la categoría de estado Borrador. Los equipos pueden usar esta vista para ver qué problemas aún se están definiendo o están pendientes de ser asignados y resueltos.
- La vista de problemas activos es una vista general de todos los problemas en los que se está trabajando actualmente. En esta vista se mostrará cualquier problema con un estado en las categorías de estado No iniciado, Iniciado o Finalizado.
- La vista de todos los problemas es una vista de cuadrícula que muestra todos los problemas del proyecto, tanto los borradores como los problemas activos.
- La vista archivada muestra todos los problemas archivados.

¿Está buscando un problema

Puedes encontrar un problema buscando parámetros específicos. Para obtener más información sobre cómo refinar la búsqueda, consulte [Buscar en CodeCatalyst](#).

Para buscar un problema

1. Vaya a su proyecto.

2. Usa la barra de búsqueda para buscar problemas o información relacionada con ellos. Puede utilizar los parámetros de consulta para afinar la búsqueda. Para obtener más información, consulte [Buscar en CodeCatalyst](#).

Problemas de clasificación

De forma predeterminada, los problemas CodeCatalyst se ordenan por orden manual. El orden manual muestra los problemas en el orden al que los trasladan los usuarios. Puede arrastrar y soltar los problemas ordenados de forma manual para cambiar su orden. Esta opción de clasificación es útil para eliminar la acumulación de problemas pendientes y priorizar los problemas.

La siguiente tabla muestra cómo se pueden ordenar los problemas tanto en la vista de cuadrícula como en la de tablero.

Opciones de clasificación de la vista de cuadrícula	Opciones de clasificación de vistas de tablero
Pedido manual	Pedido manual
Última actualización	Última actualización
Priority (Prioridad)	Priority (Prioridad)
Estimación	Estimar
Título	Título
ID	
Status	
Bloqueada	
Campos personalizados	

Utilice el siguiente procedimiento para cambiar la forma en que se ordenan los problemas.

Para ordenar los problemas

1. Vaya a su proyecto.
2. En el panel de navegación, elija Problemas. La vista predeterminada es el tablero.
3. (Opcional) Seleccione Temas activos para abrir el menú desplegable del conmutador de vistas de temas y acceder a otra vista de temas.
4. Para ordenar una vista de cuadrícula, hay dos opciones:
 - a. Elija el encabezado del campo por el que desee ordenar. Al elegir el encabezado, se alternará entre orden ascendente y descendente.
 - b. Seleccione el menú desplegable Ordenar por y elija un parámetro por el que ordenar. Los problemas se ordenarán en orden ascendente.
5. Para ordenar una vista del tablero, seleccione el menú desplegable Ordenar por y elija un parámetro por el que ordenar. Las cuestiones se ordenarán en orden ascendente.

Problemas de agrupamiento

La agrupación se utiliza para organizar las cuestiones en el foro según varios parámetros, como el responsable, las etiquetas y la prioridad.

Para agrupar asuntos

1. Vaya a su proyecto.
2. En el panel de navegación, elija Problemas. La vista predeterminada es el tablero.
3. (Opcional) Seleccione Temas activos para abrir el menú desplegable del conmutador de vistas de temas y acceder a otra vista de temas.
4. Seleccione Grupo.
5. En Agrupar por, elija un parámetro por el que agrupar:
 - Si elige Asignatario o Prioridad, elija el orden de grupo.
 - Si eliges Etiqueta, seleccione las etiquetas y, a continuación, seleccione Orden grupal.
6. (Opcional) Seleccione la opción Mostrar grupos vacíos para mostrar u ocultar los grupos que actualmente no tienen ningún problema asignado.
7. La vista se actualiza a medida que hace sus elecciones. Un problema solo aparece en el grupo que coincide con los parámetros configurados.

Problemas de filtrado

Utilice el filtrado para buscar problemas que contengan un nombre, una prioridad, una etiqueta, campos personalizados o un responsable específicos.

Para filtrar problemas

1. Vaya a su proyecto.
2. En el panel de navegación, elija Problemas.
3. (Opcional) Selecciona Problemas activos para abrir el menú desplegable del conmutador de consulta de problemas y acceder a otra vista de problemas.

Note

Para filtrar según una cadena del nombre o la descripción del problema, introduce la cadena en la barra de búsqueda de problemas.

4. Selecciona Filtro y, a continuación, selecciona + Añadir filtro.
5. Elija los parámetros por los que desee filtrar. Puede elegir varios filtros y parámetros. Puede configurar los filtros para mostrar los problemas que coincidan con todos los filtros o con cualquier filtro individual seleccionando y o. La vista se actualizará para mostrar los problemas que coinciden con el filtro.

Crear una vista de problemas

Puede crear [vistas](#) para ver rápidamente los problemas que coincidan con un conjunto concreto de filtros. Esto puede ayudarle a ahorrar tiempo y a ver rápidamente los problemas por los que anteriormente había filtrado, agrupado u ordenado.

Para crear una vista de problemas

1. En el panel de navegación, elija Problemas.
2. (Opcional) Según su caso de uso, es posible que desee crear una vista a partir de una vista existente. Para ir a otra vista, selecciona Cuestiones activas para abrir el menú desplegable del conmutador de vistas de incidencias y selecciona la vista.

3. (Opcional) Configura los filtros, las agrupaciones y la clasificación antes de crear la vista. Puede añadirlos al crear una vista, pero si lo hace antes, podrá obtener una vista previa de lo que se muestra en la vista antes de crearla.
4. Abre el menú desplegable de la vista de problemas desde la barra de encabezado. Para crear una vista de tablero en la que las cuestiones se visualicen en columnas según su estado, selecciona el signo + en la columna Tablero. Para crear una vista de cuadrícula en la que las cuestiones se vean en una lista, selecciona el signo + en la columna de cuadrícula. Puede cambiar el tipo de vista antes de crearla si cambia de opinión.
5. En el cuadro de diálogo Crear vista, introduzca un nombre para la vista.
6. Los campos Filtrar, Agrupar problemas por y Ordenar problemas por se rellenan en función de la configuración de la vista actual. Actualícelos si es necesario.
7. Seleccione Crear vista para crear la vista y cambiar a ella.

Problemas de exportación

Puede exportar los problemas de la vista actual a un archivo.xlsx. Para exportar los problemas, lleve a cabo los siguientes pasos.

Para exportar números

1. Vaya a su proyecto.
2. En la barra de navegación, selecciona Problemas.
3. Selecciona Cuestiones activas para abrir el menú desplegable del conmutador de vistas de incidencias y navega hasta la vista que contiene las incidencias que deseas exportar. Solo se exportarán los números que se muestran en la vista.
4. Seleccione el menú de puntos suspensivos y elija Exportar a Excel.
5. Se descarga el archivo.xlsx. De forma predeterminada, se titula el nombre del proyecto y la fecha en que se completó la exportación.

Configuración de los ajustes de problemas

En los siguientes temas se detalla cómo configurar los ajustes para los problemas de CodeCatalyst.

Temas

- [Habilitar o deshabilitar varios cesionarios](#)

- [Configurar la estimación del esfuerzo del problema](#)
- [Estados](#)
- [Etiquetas](#)
- [Campos personalizados](#)
- [Visualización y administración de los archivos adjuntos](#)

Habilitar o deshabilitar varios cesionarios

Siga estos pasos para configurar la configuración para varios cesionarios en caso de problemas en CodeCatalyst

Para activar o desactivar varios cesionarios

1. En el panel de navegación, selecciona Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.
3. En Asignatario, en la sección Configuración básica, activa el indicador para permitir que se asignen varios responsables al mismo problema. Un problema puede tener hasta 10 personas asignadas. Si no habilitas esta opción, solo podrás asignar un responsable a una emisión.

Configurar la estimación del esfuerzo del problema

Siga estos pasos para configurar la configuración de las estimaciones del esfuerzo en CodeCatalyst caso de problemas relacionados con.

Para configurar la estimación del esfuerzo en caso de problemas

1. En el panel de navegación, elija Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.
3. En Estimación, en la sección Configuración básica, elige cómo se mostrarán los valores de estimación. Los tipos de estimaciones disponibles son el tamaño de una camiseta, la secuenciación de Fibonacci o las estimaciones de Hide. Cuando se actualice el tipo de estimación, no se perderá ningún dato y el valor estimado de todos los números se convertirá automáticamente. El mapeo de conversión se muestra en la siguiente tabla.

Talla de camiseta	Secuencia de Fibonacci
XS	1
XS	2
S	3
M	5
L	8
XL	13

Estados

Puedes añadir estados personalizados a tu tablero. Cada estado personalizado debe pertenecer a una de las siguientes categorías: Borrador, No iniciado, Iniciado o Finalizado. Las categorías de estado se utilizan para ayudar a organizar los estados y completar las vistas predeterminadas. Para obtener más información sobre los estados y las categorías de estado, consulte [Estado y categorías de estado](#) y para obtener más información sobre las vistas, consulte [Problemas de búsqueda y visualización](#)

Para crear un estado

1. En el panel de navegación, elija Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.
3. En Estados, selecciona el icono con el signo más situado junto a la categoría en la que quieres que aparezca el estado.
4. Asigne un nombre al estado y, a continuación, seleccione el icono de marca de verificación.

Note

Seleccione el icono X para cancelar la adición de un estado.

El estado personalizado ahora está visible en tu tablero y se muestra como una opción al crear una incidencia.

Para editar un estado

1. En el panel de navegación, selecciona Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.
3. En Estados, selecciona el icono de edición situado junto al estado que deseas editar o cambiar.
4. Edita el estado y, a continuación, selecciona el icono de marca de verificación.

El estado editado ahora está visible en tu tablero.

Para mover un estado

1. En el panel de navegación, elija Problemas.
2. Seleccione el icono de puntos suspensivos y seleccione Configuración.
3. En Estados, elige el estado que quieras mover.
4. Arrastra y suelta el estado donde quieras que esté.

Note


Solo puede mover un estado dentro de su categoría designada.

Los estados ahora están reordenados en tu tablero.

Para desactivar un estado


1. En el panel de navegación, seleccione Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.
3. En Estados, elige el estado que quieras desactivar.

4. En el estado que deseas desactivar, selecciona la opción para activarlo. El estado ahora está atenuado.

 Note

El estado desactivado aparece en el tablero hasta que se eliminen todos los problemas. Los problemas no se pueden añadir a un estado desactivado.

5. Para reactivar un estado desactivado, selecciona la opción para activar el estado. El estado ya no aparece atenuado.

 Note

Debe haber al menos un estado activo en cada categoría. Si solo hay un estado en la categoría, no puede desactivarlo.

Etiquetas

Puede personalizar las etiquetas en función de los problemas. Esto incluye editar la etiqueta y cambiar el color. Las etiquetas pueden ayudarte a clasificar y organizar tu trabajo.

Crear una etiqueta

En CodeCatalyst, puede crear etiquetas añadiéndolas al crear una nueva edición o al editar una edición existente. Para obtener más información, consulte [Crear un problema en CodeCatalyst](#) y [Edición y colaboración en temas de CodeCatalyst](#).

Edición de una etiqueta

Utilice el siguiente procedimiento para cambiar el nombre o el color de una etiqueta existente.

Para editar una etiqueta

1. En el panel de navegación, elija Problemas.
2. Selecciona Problemas activos para abrir el menú desplegable del conmutador de visualización de problemas y selecciona Configuración.

3. En el mosaico Etiquetas hay una lista de las etiquetas utilizadas en el proyecto. Elija el icono de edición situado junto a la etiqueta que desee editar. Realice una o más de las siguientes acciones:
 - a. Edita el nombre de la etiqueta.
 - b. Para cambiar el color, elija la rueda de colores. Usa el selector para elegir un color nuevo.
4. Para guardar los cambios realizados en la etiqueta, selecciona el icono de marca de verificación.
5. La etiqueta modificada ahora está visible en la lista de etiquetas disponibles. También puedes ver cuántos números utilizan esa etiqueta.

Note

Puede elegir el número que aparece junto a cada etiqueta para ir a la página Todos los números y ver todos los números que contienen esa etiqueta.

Eliminar una etiqueta

Actualmente no puedes eliminar una etiqueta de problemas en CodeCatalyst. Si eliminas una etiqueta de todos los números, la etiqueta aparecerá en la sección Etiquetas no utilizadas de la sección Etiquetas de la configuración del número. Las etiquetas no utilizadas aparecen al final de la lista de etiquetas cuando se utilizan filtros o se añaden etiquetas a una edición. Puedes encontrar un resumen de todas las etiquetas (usadas y no utilizadas) y de las ediciones que las contienen en la configuración de ediciones.

Campos personalizados

Puede crear campos personalizados para ayudar a organizar y ver el trabajo de su proyecto. Los campos personalizados se añaden a la lista de filtros disponibles en Filter para que puedas filtrar los problemas por campos personalizados. Los campos personalizados son pares de nombre y valor. Se filtra por el nombre del campo personalizado y, a continuación, por el valor de ese campo personalizado.

Un problema puede tener varios campos personalizados.

Crear un campo personalizado

En CodeCatalyst, puede crear campos personalizados agregándolos al crear un problema o al editar un problema existente. Para obtener más información, consulte [Crear un problema en CodeCatalyst](#) y [Edición y colaboración en temas de CodeCatalyst](#).

Eliminar un campo personalizado

Si eliminas un campo personalizado, debes eliminar el campo personalizado de cada asunto al que se añade. Cuando se elimina un campo personalizado, ya no lo verás en Filter. Puedes usar filtros para ver todos los problemas con un campo personalizado y eliminarlos editando los problemas. Para obtener más información, consulte [Problemas de búsqueda y visualización](#) y [Edición de una edición](#).

Visualización y administración de los archivos adjuntos

Puedes ver una tabla con todos los archivos adjuntos añadidos a las incidencias de tu proyecto en la configuración de incidencias. Esta tabla incluye detalles de cada archivo adjunto, incluida información como el tipo de contenido, cuándo se agregó, el número al que se agregó y su estado, y el tamaño del archivo.

Esta tabla se puede utilizar para identificar fácilmente los archivos adjuntos de gran tamaño que estén rellenos o archivados y eliminarlos y así liberar espacio de almacenamiento.

Important

Amazon no escanea ni analiza los archivos adjuntos a los números CodeCatalyst. Cualquier usuario puede añadir un archivo adjunto a una publicación que pueda contener código o contenido malintencionado. Asegúrese de que los usuarios conozcan las mejores prácticas en lo que respecta a la administración de los archivos adjuntos y a la protección contra códigos, contenidos o virus maliciosos.

Para ver y gestionar todos los archivos adjuntos de un proyecto

1. En el panel de navegación, seleccione Problemas.
2. Seleccione el icono de puntos suspensivos y seleccione Configuración.
3. Seleccione la pestaña Adjuntos.

Cuotas para emisiones en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites para las emisiones en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Recurso	Cuota predeterminada
Problemas activos	Máximo de 1000 por proyecto.
Tamaño del archivo adjunto	Máximo 500 MB por archivo adjunto. El almacenamiento total máximo de archivos adjuntos se ve afectado por los límites generales de almacenamiento de su espacio. Para obtener más información, consulte Precios .
Número total de problemas (activos y archivados)	Máximo de 100 000 por proyecto.
Vistas guardadas	Máximo de 50 vistas de problemas guardadas por proyecto.
Número de solicitudes de cambios que puedes vincular a una edición	Máximo de 50 solicitudes de cambios por publicación.
Estados (por proyecto)	Máximo de 50 por proyecto.
Estados (por número)	Máximo de 50 por número.
Etiquetas (por proyecto)	Máximo de 200 por proyecto.
Etiquetas (por número)	Máximo de 50 por número.
Campos personalizados (por número)	Máximo de 50 por número.
Cesionarios	Máximo de 10 por número.
Comentarios	Máximo de 1000 por número.

Recurso	Cuota predeterminada
Tareas	Máximo de 100 por número.

Identidad, permisos y acceso en CodeCatalyst

Cuando inicias sesión en Amazon CodeCatalyst por primera vez, creas un AWS Builder ID. AWS Los ID de constructor no existen en AWS Identity and Access Management. El nombre de usuario que elija al iniciar sesión por primera vez se convierte en el seudónimo exclusivo de su identidad.

En CodeCatalyst, puedes iniciar sesión por primera vez de dos maneras:

- Como parte de la creación de un espacio.
- Como parte de la aceptación de una invitación a un proyecto o espacio en CodeCatalyst.

El rol o los roles asociados a tu identidad determinan las acciones que puedes realizar CodeCatalyst. Los roles del proyecto, como administrador del proyecto y colaborador, son específicos de un proyecto, por lo que puede tener un rol en un proyecto y otro diferente en otro proyecto. Si crea un espacio, CodeCatalyst se le asigna automáticamente el rol de administrador del espacio. Cuando los usuarios aceptan invitaciones a un proyecto, CodeCatalyst añaden esas identidades al espacio y les asigna la función de acceso limitado. Cuando invitas a los usuarios a los proyectos, eliges el rol que quieres que tengan en el proyecto, lo que determina qué acciones pueden y qué no pueden realizar dentro del proyecto. La mayoría de los usuarios que trabajan en un proyecto solo necesitan el rol de colaborador para realizar sus tareas. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

Además de una función de proyecto, los usuarios de un proyecto necesitan un token de acceso personal (PAT) para acceder a los repositorios de origen de un proyecto cuando utilizan clientes Git o entornos de desarrollo integrados (IDE). Los miembros del proyecto pueden usar esta PAT con aplicaciones de terceros como una contraseña específica de la aplicación asociada a su identidad. CodeCatalyst Por ejemplo, al clonar un repositorio de origen en un equipo local, debe proporcionar una PAT y su nombre de CodeCatalyst usuario.

Puede configurar el acceso entre AWS los recursos CodeCatalyst y los recursos mediante un [rol de servicio](#) para realizar acciones como acceder a las AWS CloudFormation pilas y los recursos al implementar acciones en los flujos de trabajo. Debe configurar el acceso entre AWS los recursos CodeCatalyst y los recursos para que se ejecuten las acciones del flujo de trabajo que se incluyen en las plantillas del proyecto.

Temas

- [Trabajar con roles en Amazon CodeCatalyst](#)

- [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#)
- [Autenticación multifactorial \(MFA\) en Amazon CodeCatalyst](#)
- [Seguridad en Amazon CodeCatalyst](#)
- [Monitorización en Amazon CodeCatalyst](#)
- [Cuotas de identidad, permisos y acceso en CodeCatalyst](#)
- [Solución de problemas](#)

Trabajar con roles en Amazon CodeCatalyst

En Amazon CodeCatalyst, puedes asignar funciones a los usuarios tanto a nivel de proyecto como a nivel de espacio. En un proyecto, un rol especifica lo que un usuario puede hacer en un proyecto con los recursos de ese proyecto. Los usuarios se convierten en miembros de un espacio cuando se unen a un proyecto. Puede añadir o eliminar usuarios como administradores de un espacio. El rol de administrador del espacio tiene los permisos más amplios de todos los roles. CodeCatalyst Como práctica recomendada, asigne a los usuarios los permisos más limitados necesarios para realizar sus tareas.

Puede asignar funciones a los usuarios del espacio. También puede asignar funciones a los usuarios en los proyectos de los que son miembros. Cada usuario solo puede tener un rol en un proyecto o espacio, pero los usuarios pueden tener roles diferentes en cada proyecto y espacio. Por ejemplo, un usuario puede tener el rol de administrador del proyecto en un proyecto y el rol de colaborador en otro proyecto.

Temas

- [Tipos de roles](#)
- [Permisos disponibles para cada función](#)
- [Ver y cambiar las funciones de los usuarios](#)

Tipos de roles

Hay tres funciones disponibles para un espacio:

- Administrador del espacio
- Usuario avanzado
- Acceso limitado

A los usuarios que aceptan una invitación a un proyecto se les asigna automáticamente el rol de acceso limitado en el espacio que contiene el proyecto.

Hay cuatro roles disponibles para los miembros de un proyecto:

- Administrador del proyecto
- Colaborador
- Revisor
- Solo lectura

Al añadir un usuario a un proyecto, CodeCatalyst se le asigna automáticamente el rol de acceso limitado. Si eliminas un usuario de todos los proyectos, elimina CodeCatalyst automáticamente el rol de acceso limitado de ese usuario.

Función de administrador de espacios

El rol de administrador de espacios es el rol más poderoso en CodeCatalyst. Asigne la función de administrador del espacio únicamente a los usuarios que necesiten administrar todos los aspectos de un espacio, ya que esta función tiene todos los permisos necesarios CodeCatalyst. Los usuarios con el rol de administrador del espacio son los únicos usuarios que pueden añadir o eliminar a otros usuarios del rol de administrador del espacio y eliminar el espacio.

Al crear un espacio, CodeCatalyst se le asigna automáticamente el rol de administrador del espacio. Como práctica recomendada, le recomendamos que añada este rol al menos a otro usuario que pueda desempeñarlo en caso de que el creador del espacio original no esté disponible.

Rol de usuario avanzado

El rol de usuario avanzado es el segundo rol más importante en CodeCatalyst los espacios, pero no tiene acceso a los proyectos de un espacio. Está diseñado para usuarios que necesitan poder crear proyectos en un espacio y ayudar a gestionar los usuarios y los recursos del espacio. Asigne la función de usuario avanzado a los usuarios que son líderes de equipo o gerentes y que necesitan la capacidad de crear proyectos y administrar los usuarios del espacio como parte de su trabajo.

Función de acceso limitado

El rol de acceso limitado es el rol que tendrán la mayoría de los usuarios en CodeCatalyst los espacios. Es el rol que se asigna automáticamente a los usuarios cuando aceptan una invitación a un proyecto en un espacio. Proporciona los permisos limitados que necesitan para trabajar en el

espacio que contiene ese proyecto. Asigna la función de acceso limitado a los usuarios a los que invites directamente al espacio, a menos que su trabajo requiera que administren algún aspecto del espacio.

Función de administrador del proyecto

El rol de administrador de proyectos es el rol más poderoso de un CodeCatalyst proyecto. Asigne esta función únicamente a los usuarios que necesiten administrar todos los aspectos de un proyecto, incluida la edición de la configuración del proyecto, la administración de los permisos del proyecto y la eliminación de proyectos.

Los roles del proyecto no tienen ningún permiso a nivel de espacio. Por lo tanto, los usuarios con el rol de administrador de proyectos no pueden crear proyectos adicionales. Solo los usuarios con el rol de administrador de Space o de usuario avanzado pueden crear proyectos.

Note

El rol de administrador del espacio tiene todos los permisos CodeCatalyst.

Función de colaborador

El rol de colaborador está destinado a la mayoría de los miembros de un CodeCatalyst proyecto. Asigne este rol a los usuarios que necesiten poder trabajar con el código, los flujos de trabajo, los problemas y las acciones de un proyecto.

Función de revisor

El rol de revisor está pensado para los usuarios que necesitan poder interactuar con los recursos de un proyecto, como las solicitudes de cambios y los problemas, pero no deben crear y combinar código, crear flujos de trabajo ni iniciar o detener las ejecuciones de flujos de trabajo en un CodeCatalyst proyecto. Asigna el rol de revisor a los usuarios que necesitan poder aprobar y comentar las solicitudes de incorporación de datos, crear, actualizar, resolver problemas y comentarlos, y ver el código y los flujos de trabajo de un proyecto.

Función de solo lectura

La función de solo lectura está destinada a los usuarios que necesitan ver los recursos y el estado de los recursos, pero no interactuar con ellos ni contribuir directamente al proyecto. Los usuarios con este rol no pueden crear recursos CodeCatalyst, pero pueden verlos y copiarlos, por ejemplo,

clonando repositorios y descargando archivos adjuntos a los problemas a un ordenador local. Asigne la función de solo lectura a los usuarios que necesiten ver los recursos y el estado del proyecto, pero no interactuar directamente con él.















Permisos disponibles para cada función


En la siguiente tabla se muestran los permisos disponibles para cada CodeCatalyst rol. Utilice los enlaces para ir al conjunto de permisos correspondiente.

- [Space permissions](#)
- [Extensions permissions](#)
- [Project permissions](#)
- [Source repository permissions](#)
- [Dev Environment permissions](#)
- [Package repository and package permissions](#)
- [Workflow permissions](#)
- [Issues permissions](#)
- [Custom blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
---------	--------------------------------------	-----------------------------	----------------------------	---------------------------------------	------------------------	--------------------	-------------------------






















Permisos de espacio

Crea espacio							
Edita los detalles de							










































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
facturación del espacio							
Configura y habilita el inicio de sesión único							
Eliminar el inicio de sesión único							
Habilite las funciones generativas de IA para un espacio							
Deshabilita las funciones de IA generativa para un espacio							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Eliminar un espacio							
Añada otros usuarios a la función de administrador del espacio							
Elimine a otros usuarios de la función de administrador del espacio							
Crea un equipo							
Eliminar equipo							
Actualizar equipo							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Desactiva los recursos de la máquina para el espacio							
Habilite los recursos de la máquina para el espacio							
Crea un proyecto							
Asocie una conexión de cuenta de AWS al espacio							





























Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Actualizar la conexión de una cuenta de AWS							
Desasociar una conexión de cuenta de AWS del espacio							
Eliminar la conexión de una cuenta de AWS y eliminarla del espacio							









Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Invita a otras personas al espacio							
Crear una conexión de VPC							
Editar conexión de VPC							
Eliminar conexión de VPC							
Vea los registros de actividad en el espacio							
Ver las conexiones de cuentas de AWS							











































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vea los incidentes de CodeCatalyst							
Ver espacio							
Ver equipos							
Ver conexiones de VPC							
Permisos de extensiones	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Instalar extensiones							
Actualizar extensiones							


















































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Eliminar extensiones							
Conectar una GitHub cuenta							
Desconectar una GitHub cuenta							
Conectar un sitio de Jira							
Desconectar un sitio de Jira							

















































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Consulta los detalles de configuración de las extensiones instaladas							
Ver las extensiones							
Permisos del proyecto	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Edita la configuración del proyecto							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Deshabilite los recursos de la máquina para el proyecto							
Habilite los recursos de máquina para el proyecto							
Eliminar el proyecto							
Invitar a los usuarios a un proyecto							













Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Cambie las funciones de los usuarios en un proyecto							
Eliminar usuarios de un proyecto							
Añadir un equipo a un proyecto							
Eliminar un equipo de un proyecto							
Cambiar el rol del equipo en el proyecto							
Ver proyecto							




















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Ver la actividad del proyecto							
Ver los equipos del proyecto							
Ver planos							
Permisos del repositorio de origen	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crea repositorios							
Enlaza repositorios							
Desvincular repositorios							





Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Eliminar repositorios							
Edite la configuración del repositorio							
Ver repositorios							
Ver la configuración del repositorio							
Clonar repositorios							
Crea ramas							
Crea reglas de sucursal							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Cambia la rama predeterminada							
Eliminar ramas							
Fusión de ramificaciones							
Actualiza las reglas de las sucursales							
Ver sucursales							
Ver las reglas de las sucursales							
Crea carpetas							


Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Eliminar carpetas							
Editar carpetas							
Ver carpetas							
Crear archivos							
Eliminar archivos							
Editar archivos							
Ver archivos							
Crea y envía confirmaciones							
Ver confirmaciones							










Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crea solicitudes de cambios							
Crea reglas de aprobación para las solicitudes de extracción							
Anule los requisitos de fusión para las solicitudes de extracción							
Actualiza las solicitudes de extracción							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Actualiza las reglas de aprobación de las solicitudes de extracción							
Visualización de solicitudes de extracción							
Consulta las reglas de aprobación de las solicitudes de extracción							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Cierre las solicitudes de extracción							
Aprueba las solicitudes de extracción							
Comenta las solicitudes de extracción							
Interactúa con Amazon Q en los comentarios sobre las solicitudes de cambios							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crear una revisión para una solicitud de cambios creada por Amazon Q							
Vincula una edición a una solicitud de extracción							
Desvincular una edición de una solicitud de extracción							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Permisos del entorno de desarrollo	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crea tu propio entorno de desarrollo							
Detenga su propio entorno de desarrollo							
Detenga los entornos de desarrollo creados por otros usuarios							




Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Reanude su propio entorno de desarrollo							
Vea sus propios entornos de desarrollo							
Vea los entornos de desarrollo creados por otros usuarios							
Edite su propio entorno de desarrollo							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Edite los entornos de desarrollo creados por otros usuarios							
Elimine su propio entorno de desarrollo							
Elimine los entornos de desarrollo creados por otros usuarios							


Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Cree un archivo de desarrollo para entornos de desarrollo							
Edite un archivo de desarrollo para entornos de desarrollo							
Elimine un archivo de desarrollo para entornos de desarrollo							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vea un archivo de desarrollo para entornos de desarrollo							
Repositorio de paquetes y permisos de paquetes	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crear un repositorio de paquetes							
Ver repositorios de paquetes							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Editar el repositorio de paquetes							
Eliminar el repositorio de paquetes							
Cree un repositorio de paquetes de puerta de enlace							
Vea los repositorios de paquetes de gateway							





























Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Eliminar el repositorio de paquetes de gateway							
Agregue un repositorio de paquetes ascendente							
Edite el orden de búsqueda de los repositorios ascendentes							
Elimine el repositorio de paquetes anterior							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Conectarse a un repositorio de paquetes							
Lea los paquetes de un repositorio de paquetes							
Publica paquetes en un repositorio de paquetes							
Lea y conserve los paquetes de un repositorio anterior							
Ver paquetes							





Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vea las versiones de los paquetes							
Vea los activos de la versión del paquete							
Enumere las dependencias de las versiones del paquete							
Actualización del estado de la versión del paquete							




































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Actualice la configuración de origen del paquete							
Eliminar la versión del paquete							
permisos de flujo de trabajo	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crear flujo de trabajo							
Actualizar el flujo de trabajo							
Eliminar flujo de trabajo							

















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Iniciar flujo de trabajo							
Detenga el flujo							
Cree secretos de flujo de trabajo							
Actualizar los secretos del flujo de trabajo							
Eliminar los secretos del flujo de trabajo							
Crear entornos							
Eliminar entornos							











































Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crea una flota							
Actualiza la flota							
Eliminar flota							
Administra los recursos del flujo de trabajo en otras cuentas							
Asociar una conexión de VPC a un entorno							
Desasociar una conexión de VPC con un entorno							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Asocie un entorno conectado a VPC a un flujo de trabajo							
Disociar un entorno conectado a VPC de un flujo de trabajo							
Realice un seguimiento de una confirmación en un flujo de trabajo							
Vea los entornos							







Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vea los registros de acciones de compilación							
Ver flotas							
Vea los registros de acciones de prueba							
Ver flujo de trabajo							
Ver las ejecuciones del flujo de trabajo							





















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vea los resultados de ejecución del flujo de trabajo							
Vea los secretos del flujo de trabajo							
Emite permisos	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crear problema							
Problema de actualización							
Ver problemas							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Archivar un problema							
Asignar un problema a Amazon Q							
Interactúa con Amazon Q en los comentarios sobre un problema							
Anular la asignación de Amazon Q de un problema							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Problemas de actualización creados por otros usuarios							
Ver los comentarios sobre un problema							
Crea un comentario sobre un tema							
Actualizar un comentario sobre un problema							
Crea una etiqueta							
Actualizar una etiqueta							






















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Ver etiquetas							
Añadir una etiqueta a un número							
Eliminar una etiqueta de un número							
Crea un estado personalizado para los problemas							
Actualiza un estado personalizado							
Ver un estado personalizado							





























Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Mover un estado personalizado							
Desactivar un estado personalizado							
Añadir un archivo adjunto a una edición							
Ver el archivo adjunto de un problema							
Eliminar un archivo adjunto de una edición							





























Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Vincula una solicitud de incorporación de cambios a una edición							
Desvincula a una solicitud de cambios de una edición							
Vincula un proyecto de Jira							
Desvincular un proyecto de Jira							





























Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Permisos de blueprint personalizados	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Crea un proyecto de plano personalizado							
Publique una vista previa de un plano personalizado							
Anule la publicación de una vista previa de un blueprint personalizado							















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Publica un blueprint personalizado							
Anule la publicación de un blueprint personalizado							
Añada un plano personalizado a un catálogo de planos de espacios							















Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Elimine un plano personalizado de un catálogo de planos de espacio							
Gestione los permisos de publicación de un plano personalizado							
Administre la versión del catálogo para un esquema personalizado							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Actualice un esquema personalizado							
Eliminar una versión de blueprint personalizada							
Eliminar un blueprint personalizado							
Aplicar un esquema personalizado a un proyecto							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Desasocie un esquema personalizado de un proyecto							
Actualice la versión de un blueprint personalizado aplicado							
Edite el alias de un blueprint personalizado							
Vea los planos personalizados publicados							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Permisos de notificaciones	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Configure el canal de notificaciones							
Eliminar el canal de notificaciones							
Edita la configuración de notificaciones							
Ver la configuración de notificaciones							

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Reciba automáticamente notificaciones sobre CodeCatalyst incidentes							
Configure las notificaciones por correo electrónico para su cuenta de correo electrónico asociada							
Permisos de búsqueda	Función de administrador de espacios	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura

Permiso	Función de administrador del espacio	Función de usuario avanzado	Función de acceso limitado	Función de administrador del proyecto	Función de colaborador	Función de revisor	Función de solo lectura
Busca dentro de un proyecto							
Busca en todo el espacio							

Ver y cambiar las funciones de los usuarios

Puede ver el rol asignado a un usuario. Esto le ayuda a entender qué acciones pueden realizar en un proyecto. También puedes cambiar su función si necesitan permisos adicionales.

Para ver el rol de un usuario en un proyecto

1. Navegue hasta el proyecto en el que desee ver las funciones asociadas a cada miembro del proyecto.

Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.

2. En el panel de navegación, selecciona Configuración del proyecto.
3. En la pestaña Miembros, el rol de cada miembro del proyecto se muestra en Rol.

Para cambiar las funciones de los usuarios en un proyecto

1. Navegue hasta el proyecto en el que desee cambiar los roles asociados a los miembros del proyecto.


 Tip

Puedes elegir qué proyecto quieres ver en la barra de navegación superior.

2. En el panel de navegación, selecciona Configuración del proyecto.
3. En la pestaña Miembros, en Miembros del proyecto, elija los usuarios cuyos roles desee cambiar. Elija Acción y, a continuación, elija Editar rol.
4. En Función, elija la función del proyecto y, a continuación, elija Confirmar.

Ver y cambiar los roles en el espacio

Todos los usuarios que aceptan invitaciones a un proyecto pasan a CodeCatalyst ser miembros del espacio del proyecto. Puede ver la lista de miembros del espacio. Puedes cambiar las funciones de los usuarios de Acceso limitado a Administrador del espacio para administrar mejor tu espacio y sus recursos. El rol de administrador de Space es el único rol que permite a los usuarios crear proyectos CodeCatalyst.

 Warning

El rol de administrador de Space es el rol más poderoso que existe CodeCatalyst. Los usuarios con este rol pueden realizar cualquier acción CodeCatalyst, incluida la eliminación del espacio. Asigna este rol únicamente a los usuarios que requieran este nivel de acceso a tu espacio. Para obtener más información, consulte [Función de administrador de espacios](#).

Para cambiar el rol de un usuario en el espacio

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta el espacio.

 Tip

Si pertenece a más de un espacio, puede elegir qué espacio ver en la barra de navegación superior.

3. Seleccione la pestaña Miembros.

4. Elija el usuario cuyo rol desee cambiar y, a continuación, elija Cambiar rol.
5. En Cambiar rol, elige el rol que deseas asignar y, a continuación, selecciona Confirmar.

Gestión de los tokens de acceso personal en Amazon CodeCatalyst

Para acceder a algunos CodeCatalyst recursos, como los repositorios de código fuente, en un equipo local con un cliente Git o un entorno de desarrollo integrado (IDE), debes introducir una contraseña específica de la aplicación. Puede crear un token de acceso personal (PAT) para usarlo con este fin específico. Las PAT que crees se asocian a tu identidad de usuario en todos los espacios y proyectos en los que se encuentren. CodeCatalyst Puede crear más de una PAT para su CodeCatalyst identidad.

Puede ver los nombres y las fechas de caducidad de las PAT que ha creado y eliminar las que ya no necesite. Solo puede copiar el secreto de PAT en el momento de su creación.

Note

De forma predeterminada, las PAT caducan en 1 año.

Creación de PAT

Las PAT están asociadas a su identidad de usuario en. CodeCatalyst Solo puede copiar un secreto PAT en el momento de crearlo.

Creación de PAT (consola)

Puede utilizar la consola para crear PAT en. CodeCatalyst

Para crear un token de acceso personal (consola)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de menú superior, elija su insignia de perfil y, a continuación, elija Mi configuración. Se abre la página CodeCatalyst Mis ajustes.


 Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

3. En Tokens de acceso personal, seleccione Crear.


Aparece la página Crear PAT.

4. En el nombre de la PAT, introduzca un nombre descriptivo para la PAT.
5. En Fecha de caducidad, mantenga la fecha predeterminada o elija el icono del calendario para seleccionar una fecha personalizada. La fecha de caducidad predeterminada es de 1 año a partir de la fecha actual.
6. Seleccione Crear.

 Tip

También puede crear este token si elige Clonar repositorio para un repositorio de origen.

7. Para copiar el secreto de la PAT, seleccione Copiar. Guarde el secreto PAT en un lugar donde pueda recuperarlo.

 Important

El secreto del PAT solo se muestra una vez. No podrá recuperarlo después de cerrar la ventana. Si no guardó el secreto de la PAT en una ubicación segura, puede crear otra.

Creación de PAT (CLI)

Puede usar la CLI para crear PAT en CodeCatalyst.

Para crear un token de acceso personal (AWS CLI)

1. En la terminal o en la línea de comandos, ejecute el create-access-token comando de la siguiente manera.

```
aws codecatalyst create-access-token
```

Si se ejecuta correctamente, el comando devuelve información sobre la PAT creada, como en el siguiente ejemplo.

```
{
  "secret": "value",
  "name": "marymajor-22222EXAMPLE",
  "expiresTime": "2024-02-04T01:56:04.402000+00:00"
}
```

2.

Solo puede ver el secreto de la PAT una vez, al crear la PAT. Si ha perdido un secreto de PAT o le preocupa que no esté almacenado de forma segura, puede crear otro.

Puede ver los PAT asociados a su cuenta de usuario mediante el AWS CLI. Solo puede ver información sobre la PAT y no el valor del secreto de la PAT en sí.

Note

Asegúrese de utilizar una versión reciente de la AWS CLI para trabajar con ella CodeCatalyst. Es posible que las versiones anteriores no contengan los CodeCatalyst comandos. Debe configurar su AWS CLI perfil antes de poder usarlo con CodeCatalyst. Para obtener más información, consulte [Configuración para usar el AWS CLI con CodeCatalyst](#).

Visualización de las PAT

Puede ver las PAT en CodeCatalyst. La lista muestra todas las PAT que ha asociado a su identidad de usuario. Su PAT está asociada a su perfil de usuario en todos los espacios y proyectos de CodeCatalyst. Las PAT caducadas no se muestran porque se eliminan una vez caducadas.

Visualización de las PAT (consola)

Puede utilizar la consola para ver las PAT asociadas a su identidad de usuario en CodeCatalyst

Para ver sus fichas de acceso personales (consola)

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de menú superior, elija su insignia de perfil y, a continuación, elija Mi configuración. Se abre la página CodeCatalyst Mis ajustes.

 Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

3. En Tokens de acceso personal, consulta los nombres y las fechas de caducidad de tus PAT actuales.

Visualización de las PAT (CLI)

Puede usar la CLI para ver las PAT asociadas a su identidad de usuario en CodeCatalyst.

Para ver sus fichas de acceso personales (AWS CLI)

- En la terminal o en la línea de comandos, ejecute el list-access-tokens comando de la siguiente manera.

```
aws codecatalyst list-access-tokens
```

Si se ejecuta correctamente, el comando devuelve información sobre las PAT asociadas a su cuenta de usuario, como en el ejemplo siguiente.

```
{
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
      "name": "marymajor-22222EXAMPLE",
      "expiresTime": "2024-02-04T01:56:04.402000+00:00"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
      "name": "marymajor-11111EXAMPLE",
      "expiresTime": "2023-03-12T01:58:40.694000+00:00"
    }
  ]
}
```



```
]
}
```

Eliminar las PAT

Puede eliminar las PAT asociadas a su identidad de usuario en. CodeCatalyst

Eliminar las PAT (consola)

Puede utilizar la consola para eliminar las PAT. CodeCatalyst

Para eliminar un token de acceso personal (consola)

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la barra de menú superior, elija su insignia de perfil y, a continuación, elija Mi configuración. Se abre la página CodeCatalyst Mis ajustes.

Tip

También puedes encontrar tu perfil de usuario si vas a la página de miembros de un proyecto o espacio y eliges tu nombre en la lista de miembros.

3. En Tokens de acceso personal, selecciona el selector situado junto a la PAT que deseas eliminar y, a continuación, selecciona Eliminar.

En la PAT de eliminación: ? <name> página, para confirmar la eliminación, escriba delete en el campo de texto. Elija Eliminar.

Eliminación de PAT (CLI)

Puede eliminar una PAT asociada a su identidad de usuario mediante. AWS CLI Para ello, debe proporcionar el ID de la PAT, que puede ver mediante el delete-access-token comando.

Note

Asegúrese de utilizar una versión reciente de la AWS CLI para trabajar con ella CodeCatalyst. Es posible que las versiones anteriores no contengan los CodeCatalyst

comandos. Para obtener más información sobre el uso del AWS CLI with CodeCatalyst, consulte [Configuración para usar el AWS CLI con CodeCatalyst](#).

Para eliminar un token de acceso personal (AWS CLI)

- En la terminal o en la línea de comandos, ejecute el delete-access-token comando e indique el ID de la PAT que desee eliminar. Por ejemplo, ejecute el siguiente comando para eliminar una PAT con un ID de *123EXAMPLE*.

```
aws codecatalyst delete-access-token --id a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb
```

Si se ejecuta correctamente, este comando no devuelve ninguna respuesta.

Autenticación multifactorial (MFA) en Amazon CodeCatalyst

Ya sea que haya creado su perfil de AWS Builder ID para uso personal o profesional, le recomendamos que configure la autenticación multifactor (MFA) como otro nivel de seguridad. Recomendamos especialmente configurar la MFA si es miembro de un espacio y colabora con otras personas en proyectos. Como más de una persona puede tener acceso a un proyecto, existen más oportunidades de que se produzcan brechas de seguridad.

Cuando habilitas la MFA, debes iniciar sesión en Amazon CodeCatalyst con tu correo electrónico y contraseña. Esta parte del inicio de sesión es el primer factor, en el que utilizas algo que conoces. A continuación, inicias sesión con un código o una clave de seguridad. Este es el segundo factor, que es algo que tienes. El segundo factor podría ser un código de autenticación generado por su dispositivo móvil o al tocar o presionar una clave de seguridad conectada a su computadora. En conjunto, estos múltiples factores proporcionan una mayor seguridad al impedir el acceso no autorizado.

¿Cómo registrar un dispositivo para su uso con la autenticación multifactorial

Utilice el siguiente procedimiento en Mi perfil > Autenticación multifactor para registrar el nuevo dispositivo para la autenticación multifactor (MFA).

Note

Le recomendamos que descargue primero la aplicación de autenticación adecuada en su dispositivo antes de iniciar los pasos de este procedimiento. Para obtener una lista de las aplicaciones probadas que puede utilizar con dispositivos de MFA, consulte [Aplicaciones de autenticación](#).

Cómo registrar el dispositivo para su uso con MFA


1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Perfil de usuario. Se abre la página de CodeCatalyst perfil.
3. En la página de perfil, selecciona Administrar perfil y seguridad. Se abre la página de perfil de AWS Builder ID.
4. En la parte izquierda de la página, selecciona Seguridad.
5. En la página de autenticación multifactorial, selecciona Registrar dispositivo.
6. En la página Registrar dispositivo MFA, elija uno de los siguientes tipos de dispositivos MFA y siga las instrucciones:
 - Llave de seguridad o autenticador integrado
 1. En la página Registrar la clave de seguridad de su usuario, siga las instrucciones que le dé su navegador o plataforma.

Note

Esta experiencia varía en función del sistema operativo y el navegador, por lo que debe seguir las instrucciones que se muestran en el navegador o la plataforma. Una vez que tu dispositivo se haya registrado correctamente, tendrás la opción de asociar un nombre descriptivo al dispositivo recién inscrito. Si desea cambiarlo, seleccione Cambiar nombre, escriba el nuevo nombre y, a continuación, seleccione Guardar.


- Aplicación de autenticación

1. En la página Configurar la aplicación de autenticación, IAM Identity Center muestra la información de configuración del nuevo dispositivo de MFA, incluido un gráfico de código QR. El gráfico es una representación de la clave secreta que se puede introducir manualmente en dispositivos que no admiten códigos QR.
2. Con el dispositivo de MFA físico, haga lo siguiente:
 - a. Abra una aplicación de autenticación MFA compatible. Para obtener una lista de las aplicaciones probadas que puede utilizar con dispositivos de MFA, consulte [Aplicaciones de autenticación probada](#). Si la aplicación MFA admite varios dispositivos, elija la opción para crear un nuevo dispositivo MFA.
 - b. Determine si la aplicación de MFA admite códigos QR y, a continuación, lleve a cabo alguna de las siguientes operaciones de la página Configurar la aplicación de autenticación.
 - i. Elija Mostrar código QR y, a continuación, utilice la aplicación para escanear el código QR. Por ejemplo, puede elegir el icono de la cámara o una opción similar a Escanear código. A continuación, use la cámara del dispositivo para escanear el código.
 - ii. Seleccione Mostrar clave secreta y, a continuación, introduzca esa clave secreta en su aplicación de MFA.

 Important

Cuando configure un dispositivo MFA para AWS Builder ID, guarde una copia del código QR o la clave secreta en un lugar seguro. Esto puede ayudarle si pierde el teléfono o tiene que volver a instalar la aplicación de autenticación MFA. Si ocurre alguna de estas cosas, puede volver a configurar rápidamente la aplicación para que utilice la misma configuración de MFA.

3. En la página Configurar la aplicación de Authenticator, en Código de autenticación, escriba la contraseña de uso único que aparece actualmente en el dispositivo MFA físico.

 Important

Envíe su solicitud inmediatamente después de generar el código. Si genera el código y, a continuación, espera demasiado tiempo para enviar la solicitud, el dispositivo MFA se asociará correctamente a su perfil de AWS Builder ID, pero el dispositivo MFA no estará sincronizado. Esto ocurre porque las contraseñas

temporales de un solo uso (TOTP) caducan tras un corto periodo de tiempo. Si esto ocurre, puede volver a sincronizar el dispositivo.

4. Elija Asignar MFA. El dispositivo MFA ahora puede empezar a generar contraseñas de un solo uso y ya está listo para su uso.

Aplicaciones de autenticación

Las aplicaciones de autenticación son autenticadores de terceros basados en contraseñas de un solo uso (OTP). Los usuarios pueden usar una aplicación de autenticación instalada en su dispositivo móvil o tableta como dispositivo MFA autorizado. La aplicación de autenticación de terceros debe cumplir con el RFC 6238, que es un algoritmo TOTP (contraseña de un solo uso basado en el tiempo) basado en estándares capaz de generar códigos de autenticación de seis dígitos.

Cuando se le pida la MFA, el usuario debe introducir un código válido de su aplicación de autenticación en el cuadro de entrada que aparece. Cada dispositivo MFA asignado a un usuario debe ser único. Se pueden registrar dos aplicaciones de autenticación para un usuario determinado.

Aplicaciones de autenticación probada

Si bien cualquier aplicación compatible con TOTP funcionará con el MFA de IAM Identity Center, en la siguiente tabla se enumeran las aplicaciones de autenticación de terceros conocidas entre las que puede elegir.

Sistema operativo	Aplicación de autenticación probada
Android	Authy , Duo Mobile , Authenticator , Microsoft LastPass Authenticator , Google Authenticator
iOS	Authy , Duo Mobile , Authenticator , Microsoft LastPass Authenticator , Google Authenticator

Cómo cambiar los dispositivos MFA

Tras registrar un dispositivo MFA, puede cambiarle el nombre o eliminarlo. Recomendamos tener siempre activado al menos un dispositivo MFA para ofrecer un nivel adicional de seguridad. Puede

tener registrados hasta cinco dispositivos. Para obtener más información sobre cómo añadir más, consulte [¿Cómo registrar un dispositivo para su uso con la autenticación multifactorial.](#)

Cambiar el nombre de un dispositivo MFA

Para cambiar el nombre del dispositivo MFA

1. [Abra la CodeCatalyst consola en https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Perfil de usuario. Se abre la página de CodeCatalyst perfil.
3. En la página de perfil, selecciona Administrar perfil y seguridad. Se abre la página de perfil de AWS Builder ID.
4. Elija la autenticación multifactorial en la parte izquierda de la página. Verás que Cambiar nombre aparece atenuado cuando llegues a la página.
5. Seleccione el dispositivo MFA que desea cambiar. Elija Rename (Cambiar nombre). Luego aparece un modal.
6. En el mensaje que se abre, introduzca el nuevo nombre en el nombre del dispositivo MFA y, a continuación, elija Cambiar nombre. El dispositivo al que se le ha cambiado el nombre aparece en Dispositivos de autenticación multifactor (MFA).

Eliminar un dispositivo MFA

Para eliminar un dispositivo MFA

1. Abra la CodeCatalyst consola en [https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. En la parte superior derecha, selecciona la flecha situada junto al icono con tu primera inicial y, a continuación, selecciona Perfil de usuario. Se abre la página de CodeCatalyst perfil.
3. En la página de perfil, selecciona Administrar perfil y seguridad. Se abre la página de perfil de AWS Builder ID.
4. Elija la autenticación multifactorial en la parte izquierda de la página. Verás que Eliminar aparece atenuado cuando llegues a la página.
5. Seleccione el dispositivo MFA que desea cambiar. Elija Eliminar. Aparece un modal que dice ¿Eliminar dispositivo MFA? . Sigue las instrucciones para eliminar el dispositivo.
6. Elija Eliminar. El dispositivo eliminado ya no aparece en Dispositivos de autenticación multifactor (MFA).

Seguridad en Amazon CodeCatalyst

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de los espacios más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS 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. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de cumplimiento aplicables CodeCatalyst, consulte [AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos](#) .
- Seguridad en la nube: su responsabilidad viene determinada por los AWS servicios 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 aplicables

Esta documentación te ayuda a entender cómo aplicar el modelo de responsabilidad compartida cuando utilizas Amazon CodeCatalyst. Le muestra cómo configurarlo para CodeCatalyst cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus CodeCatalyst recursos.

Contenido

- [Protección de datos en Amazon CodeCatalyst](#)
- [Identity and Access Management y Amazon CodeCatalyst](#)
- [Validación de conformidad para Amazon CodeCatalyst](#)
- [Resiliencia en Amazon CodeCatalyst](#)
- [Seguridad de infraestructuras en Amazon CodeCatalyst](#)
- [Análisis de configuración y vulnerabilidad en Amazon CodeCatalyst](#)
- [Tus datos y privacidad en Amazon CodeCatalyst](#)
- [Mejores prácticas para las acciones de flujo de trabajo en Amazon CodeCatalyst](#)
- [CodeCatalyst modelo de confianza](#)

Protección de datos en Amazon CodeCatalyst

La seguridad y el cumplimiento son una responsabilidad compartida entre Amazon CodeCatalyst y el cliente, del AWS mismo modo que el [modelo](#) de se aplica al uso de AWS los recursos utilizados en un flujo de trabajo. Como se describe en este modelo, CodeCatalyst es responsable de proteger la infraestructura global del servicio. Es responsable de mantener el control sobre su contenido que se encuentra alojado en esta infraestructura. Este modelo de responsabilidad compartida se aplica a la protección de datos en CodeCatalyst.

Con fines de protección de datos, le recomendamos que proteja las credenciales de su cuenta y que configure la autenticación multifactorial al iniciar sesión. Para obtener más información, consulte [Autenticación multifactorial \(MFA\) en Amazon CodeCatalyst](#).

No introduzcas información confidencial o confidencial, como las direcciones de correo electrónico de tus clientes, en etiquetas o campos de formato libre, como el campo Nombre. Esto incluye los nombres de los recursos y cualquier otro identificador que introduzcas, además de los que estén conectados. CodeCatalyst Cuentas de AWS Por ejemplo, no introduzcas información confidencial o confidencial como parte de los nombres de los espacios, proyectos o flotas de despliegue. Todos los datos que introduzca en las etiquetas, los nombres o los campos de formato libre que se utilizan para los nombres se pueden utilizar para los registros de facturación o diagnóstico o se pueden incluir en las rutas URL. Esto se aplica al uso de la consola, la APIAWS CLI, el CodeCatalyst Action Development Kit o cualquier AWS SDK.

Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya ninguna información de credenciales de seguridad en la URL para validar su solicitud a ese servidor.

CodeCatalyst Los repositorios de origen se cifran automáticamente cuando están inactivos. No es necesaria ninguna acción por parte del cliente. CodeCatalyst también cifra los datos del repositorio en tránsito mediante el protocolo HTTPS.

CodeCatalyst es compatible con MFA. Para obtener más información, consulte [Autenticación multifactorial \(MFA\) en Amazon CodeCatalyst](#).

Cifrado de datos

CodeCatalyst almacena y transfiere datos de forma segura dentro del servicio. Todos los datos se cifran en tránsito y en reposo. Todos los datos creados o almacenados por el servicio, incluidos los metadatos del servicio, se almacenan de forma nativa en el servicio y se cifran.

Note

Si bien la información sobre los problemas se almacena de forma segura en el servicio, la información sobre los problemas pendientes también se almacena en la memoria caché local del navegador, donde se consultaban los tableros de temas, los trabajos pendientes y los problemas individuales. Para una seguridad óptima, asegúrate de borrar la memoria caché del navegador para eliminar esta información.

Si utilizas recursos vinculados CodeCatalyst, como una conexión de cuenta a un repositorio vinculado Cuenta de AWS o a un repositorio vinculado GitHub, los datos en tránsito desde CodeCatalyst ese recurso vinculado se cifran, pero el tratamiento de los datos en ese recurso vinculado lo gestiona ese servicio vinculado. Para obtener más información, consulta la documentación del servicio vinculado y [Mejores prácticas para las acciones de flujo de trabajo en Amazon CodeCatalyst](#).

Administración de claves

CodeCatalyst no admite la administración de claves.

Privacidad del tráfico entre redes

Al crear un espacio CodeCatalyst, se elige el Región de AWS lugar en el que se almacenarán los datos y los recursos de ese espacio. Los datos y metadatos del proyecto nunca salen de ahí Región de AWS. Sin embargo, para facilitar la navegación interna CodeCatalyst, se replica un conjunto limitado de metadatos de espacio, proyecto y usuario Regiones de AWS en toda la [partición](#). No se replicarán Regiones de AWS fuera de esa partición. Por ejemplo, si elige EE.UU. Oeste (Oregón) como espacio para crear su espacio, sus datos no se replicarán en las regiones de China o China. Región de AWS AWS GovCloud (US) Para obtener más información, consulte [Gestión Regiones de AWS](#), [infraestructura AWS global](#) y puntos [finales AWS de servicio](#).

Los datos replicados Regiones de AWS dentro de una partición incluyen:

- Un valor hash cifrado que representa el nombre del espacio para garantizar la exclusividad de los nombres de los espacios. Este valor no es legible por humanos y no expone los nombres reales de los espacios
- El identificador único del espacio
- Metadatos del espacio que ayudan a navegar por los espacios

- El Región de AWS lugar donde se encuentra el espacio
- Los identificadores únicos de todos los proyectos del espacio
- El ID de rol que indica el rol de un usuario en un espacio o proyecto
- Al registrarse CodeCatalyst, datos y metadatos sobre el proceso de registro, incluidos:
 - El identificador único del ID de creador de AWS
 - El nombre para mostrar del usuario en su ID de creador de AWS
 - El alias del usuario en su ID de creador de AWS
 - La dirección de correo electrónico utilizada cuando el usuario se registró en su ID de creador de AWS
 - El progreso del proceso de registro
 - Si creas un espacio como parte del proceso de registro, el Cuenta de AWS identificador que se utiliza como cuenta de facturación del espacio

Los nombres de los espacios son únicos en todos los espacios CodeCatalyst. Asegúrese de no incluir datos confidenciales en el nombre del espacio.

Al trabajar con recursos vinculados y cuentas conectadas, como una conexión a un repositorio Cuenta de AWS o a un GitHub repositorio, recomendamos configurar las ubicaciones de origen y destino con el nivel de seguridad más alto compatible con cada una de ellas. CodeCatalyst asegura la conexión entre Cuentas de AWS las zonas de disponibilidad y las mismas mediante el uso de Transport Layer Security (TLS) 1.2. Regiones de AWS

Identity and Access Management y Amazon CodeCatalyst

En Amazon CodeCatalyst, creas y utilizas un ID de AWS constructor para iniciar sesión y acceder a tus espacios y proyectos. Un AWS Builder ID no es una identidad en AWS Identity and Access Management (IAM) ni existe en unCuenta de AWS. Sin CodeCatalyst embargo, se integra con IAM cuando se verifica un espacio con fines de facturación y cuando se conecta a un espacio Cuenta de AWS para crear y usar recursos en él. Cuenta de AWS

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién puede estar autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos. IAM es un servicio de Servicio de AWS que se puede utilizar sin cargo adicional.

Al crear un espacio en Amazon CodeCatalyst, debes conectarlo Cuenta de AWS como cuenta de facturación de tu espacio. Debes tener permisos de administrador en el espacio Cuenta de AWS para verificar el CodeCatalyst espacio o tener el permiso. También tiene la opción de añadir una función de IAM a su espacio que CodeCatalyst podrá utilizar para crear y acceder a los recursos del espacio conectado Cuenta de AWS. Esto se denomina [rol de servicio](#). Puede optar por crear conexiones con más de una cuenta Cuenta de AWS y crear funciones de servicio para CodeCatalyst cada una de esas cuentas.

Note

La facturación CodeCatalyst se realiza en la cuenta Cuenta de AWS designada como cuenta de facturación. Sin embargo, si crea un rol de CodeCatalyst servicio en esa función Cuenta de AWS o en cualquier otra conexión Cuenta de AWS, los recursos creados y utilizados por la función de CodeCatalyst servicio se facturarán en esa función conectada Cuenta de AWS. Para obtener más información, consulta [Gestión de la facturación](#) en la Guía del CodeCatalyst administrador de Amazon.

Temas

- [Políticas basadas en la identidad en IAM](#)
- [Acciones políticas en IAM](#)
- [Recursos de políticas en IAM](#)
- [Las condiciones políticas son las claves de la IAM](#)
- [Ejemplos de políticas basadas en la identidad para las conexiones CodeCatalyst](#)
- [Uso de etiquetas para controlar el acceso a los recursos de conexión de la cuenta](#)
- [CodeCatalyst referencia de permisos](#)
- [Uso de roles vinculados a servicios de CodeCatalyst](#)
- [AWS políticas gestionadas para Amazon CodeCatalyst](#)
- [Funciones de IAM para el CodeCatalyst acceso de Amazon a los recursos AWS](#)

Políticas basadas en la identidad en IAM

Las políticas basadas en la identidad son documentos de política de permisos de JSON que se pueden adjuntar a una identidad. Esa identidad puede ser un usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en

qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades de CodeCatalyst

Para ver ejemplos de políticas CodeCatalyst basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para las conexiones CodeCatalyst](#)

Acciones políticas en IAM

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué director puede realizar qué acciones, con qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [  
    "prefix:action1",  
    "prefix:action2"  
]
```

Recursos de políticas en IAM

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué director puede realizar qué acciones, con qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). Puede

hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Las condiciones políticas son las claves de la IAM

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué director puede realizar qué acciones, con qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación lógica AND. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación OR lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Para más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para las conexiones CodeCatalyst

En CodeCatalyst, Cuentas de AWS son necesarios para gestionar la facturación de un espacio y acceder a los recursos de los flujos de trabajo del proyecto. La conexión de una cuenta se utiliza para autorizar Cuentas de AWS la adición a un espacio. En los lugares conectados se utilizan políticas basadas en la identidad. Cuentas de AWS

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de CodeCatalyst. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS

Command Line Interface (AWS CLI) o la API de AWS. Un administrador de IAM debe crear políticas de IAM que concedan a los usuarios y a los roles permiso para realizar acciones en los recursos que necesitan. El administrador debe asociar esas políticas a los usuarios que las necesiten.

El siguiente ejemplo de políticas de IAM concede permisos para acciones relacionadas con las conexiones de cuentas. Úselas para limitar el acceso a las cuentas conectadas a CodeCatalyst.

Ejemplo 1: permitir que un usuario acepte solicitudes de conexión de una sola vez Región de AWS

La siguiente política de permisos solo permite a los usuarios ver y aceptar solicitudes de conexiones entre CodeCatalyst y Cuentas de AWS. Además, la política utiliza una condición para permitir solo las acciones en la región us-west-2 y no desde otras. Regiones de AWS Para ver y aprobar la solicitud, el usuario inicia sesión en ella AWS Management Console con la misma cuenta que se especifica en la solicitud.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:GetPendingConnection"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    }
  ]
}
```

Ejemplo 2: permitir administrar las conexiones en la consola para una sola Región de AWS

La siguiente política de permisos permite a los usuarios administrar las conexiones entre CodeCatalyst y Cuentas de AWS dentro de una sola región. La política utiliza una condición para permitir las acciones únicamente en la región us-west-2 y no en otras. Regiones de AWS Tras crear una conexión, puede crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol seleccionando la opción en. AWS Management Console En la política de ejemplo, la condición de la

iam:PassRole acción incluye los principios del servicio para CodeCatalyst. Solo los roles con ese acceso se crearán en. AWS Management Console

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "codecatalyst.amazonaws.com",
            "codecatalyst-runner.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

```
]
}
```

Ejemplo 3: Denegar la administración de conexiones

La siguiente política de permisos niega a los usuarios la posibilidad de administrar las conexiones entre CodeCatalyst y Cuentas de AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Uso de etiquetas para controlar el acceso a los recursos de conexión de la cuenta

Las etiquetas se pueden adjuntar al recurso o pasar en la solicitud a los servicios que admiten el etiquetado. Los recursos de las políticas pueden tener etiquetas y algunas acciones de las políticas pueden incluirlas. Las claves de condición de etiquetado incluyen las claves de `aws:ResourceTag` condición `aws:RequestTag` y. Cuando crea una política de IAM, puede utilizar claves de condición de etiqueta para controlar lo siguiente:

- Qué usuarios pueden realizar acciones en un recurso de conexión, en función de las etiquetas que ya tiene.
- Qué etiquetas se pueden pasar en la solicitud de una acción.
- Si se pueden utilizar claves de etiqueta específicas en una solicitud.

Los siguientes ejemplos muestran cómo especificar las condiciones de las etiquetas en las políticas para los usuarios de conexiones de CodeCatalyst cuentas. Para obtener más información acerca de las claves de condición, consulte [Las condiciones políticas son las claves de la IAM](#).

Ejemplo 1: Permitir acciones basadas en las etiquetas de la solicitud

La siguiente política otorga a los usuarios permiso para aprobar las conexiones de las cuentas.

Para ello, permite las acciones `AcceptConnection` y `TagResource` si la solicitud especifica una etiqueta denominada `Project` con el valor `ProjectA`. (La clave de condición `aws:RequestTag` se utiliza para controlar qué etiquetas se pueden pasar en una solicitud de IAM). La condición `aws:TagKeys` garantiza la distinción entre mayúsculas y minúsculas de las claves de etiqueta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

Ejemplo 2: Permitir acciones basadas en etiquetas de recursos

La siguiente política otorga a los usuarios permiso para realizar acciones en los recursos de conexión de cuentas y obtener información sobre ellos.

Para ello, permite realizar acciones específicas si la conexión tiene una etiqueta denominada `Project` con ese valor `ProjectA`. (La clave de condición `aws:ResourceTag` se utiliza para controlar qué etiquetas se pueden pasar en una solicitud de IAM).

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

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "codecatalyst:GetConnection",
      "codecatalyst>DeleteConnection",
      "codecatalyst:AssociateIamRoleToConnection",
      "codecatalyst:DisassociateIamRoleFromConnection",
      "codecatalyst:ListIamRolesForConnection",
      "codecatalyst:PutBillingAuthorization"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Project": "ProjectA"
      }
    }
  }
]
}

```

CodeCatalyst referencia de permisos

En esta sección se proporciona una referencia de permisos para las acciones utilizadas con el recurso de conexión de la cuenta al Cuentas de AWS que se está conectado CodeCatalyst. En la siguiente sección, se describen las acciones relacionadas únicamente con los permisos y relacionadas con la conexión de cuentas.

Permisos necesarios para las conexiones de cuentas

Se requieren los siguientes permisos para trabajar con las conexiones de cuentas.

CodeCatalyst permisos para las conexiones de cuentas	Permisos necesarios	Recursos
AcceptConnection	Necesario para aceptar una solicitud de conexión de esta cuenta a un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	Solo admite un carácter comodín (*) en el elemento Resource de la política.

CodeCatalyst permisos para las conexiones de cuentas	Permisos necesarios	Recursos
AssociateIamRoleToConnection	Necesario para asociar un rol de IAM a una conexión de cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
DeleteConnection	Necesario para eliminar la conexión de una cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
DisassociateIamRoleFromConnection	Necesario para desasociar un rol de IAM de una conexión de cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
GetBillingAuthorization	Necesario para describir la autorización de facturación para la conexión de una cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
GetConnection	Necesario para poder conectar una cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>

CodeCatalyst permisos para las conexiones de cuentas	Permisos necesarios	Recursos
GetPendingConnection	Necesario para obtener una solicitud pendiente para conectar esta cuenta a un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	Solo admite un carácter comodín (*) en el elemento Resource de la política.
ListConnections	Necesario para enumerar las conexiones de cuentas que no están pendientes. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	Solo admite un carácter comodín (*) en el elemento Resource de la política.
ListIamRolesForConnection	Necesario para enumerar las funciones de IAM asociadas a una conexión de cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
ListTagsForResource	Necesario para enumerar las etiquetas asociadas a una conexión de cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>

CodeCatalyst permisos para las conexiones de cuentas	Permisos necesarios	Recursos
PutBillingAuthorization	Necesario para crear o actualizar la autorización de facturación de una conexión de cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
RejectConnection	Necesario para rechazar una solicitud de conexión de esta cuenta a un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	Solo admite un carácter comodín (*) en el elemento Resource de la política.
TagResource	Necesario para crear o editar las etiquetas asociadas a la conexión de una cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
UntagResource	Necesario para eliminar las etiquetas asociadas a la conexión de una cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>

Permisos necesarios para las aplicaciones del IAM Identity Center

Se requieren los siguientes permisos para trabajar con las aplicaciones del IAM Identity Center.

CodeCatalyst permisos para las aplicaciones del IAM Identity Center	Permisos necesarios	Recursos
AssociateIdentityCenterApplicationToSpace	Necesario para asociar una aplicación del centro de identidad de IAM a un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
AssociateIdentityToIdentityCenterApplication	Necesario para asociar una identidad a una solicitud del Centro de Identidad de IAM para un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
BatchAssociateIdentitiesToIdentityCenterApplication	Necesario para asociar varias identidades a una aplicación del Centro de Identidad de IAM para un CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
BatchDisassociateIdentitiesFromIdentityCenterApplication	Necesario para desasociar varias identidades de una aplicación del Centro de Identidad de IAM para un espacio. CodeCatalyst Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst permisos para las aplicaciones del IAM Identity Center	Permisos necesarios	Recursos
CreateIdentityCenterApplication	Necesario para crear una aplicación del centro de identidad de IAM. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
CreateSpaceAdminRoleAssignment	Necesario para crear una asignación de funciones de administrador para un CodeCatalyst espacio y una aplicación del IAM Identity Center determinados. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
DeleteIdentityCenterApplication	Necesario para eliminar una aplicación del IAM Identity Center. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
DisassociateIdentityCenterApplicationFromSpace	Necesario para desvincular una aplicación del Centro de Identidad de IAM de un espacio. CodeCatalyst Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst permisos para las aplicaciones del IAM Identity Center	Permisos necesarios	Recursos
DisassociateIdentityFromIdentityCenterApplication	Necesario para desasociar una identidad de una solicitud del Centro de Identidad de IAM para un espacio. CodeCatalyst Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
GetIdentityCenterApplication	Necesario para obtener información sobre una solicitud del Centro de Identidad de IAM. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
ListIdentityCenterApplications	Necesario para ver una lista de todas las aplicaciones del Centro de Identidad de ALLIAM en la cuenta. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
ListIdentityCenterApplicationsForSpace	Necesario para ver una lista de las aplicaciones del IAM Identity Center por CodeCatalyst espacio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst permisos para las aplicaciones del IAM Identity Center	Permisos necesarios	Recursos
ListSpacesForIdentityCenterApplication	Necesario para ver una lista de CodeCatalyst espacios por aplicación del IAM Identity Center. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
SynchronizIdentityCenterApplication	Necesario para sincronizar una aplicación del IAM Identity Center con el almacén de identidades de respaldo. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
UpdateIdentityCenterApplication	Necesario para actualizar una aplicación del centro de identidad de IAM. Se trata únicamente de un permiso de política de IAM, no de una acción de API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

Uso de roles vinculados a servicios de CodeCatalyst

Amazon CodeCatalyst usa roles AWS Identity and Access Management vinculados a [servicios \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. CodeCatalyst Los roles vinculados al servicio están predefinidos CodeCatalyst e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración CodeCatalyst , ya que no es necesario añadir manualmente los permisos necesarios. CodeCatalyst define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo CodeCatalyst puede asumir sus

funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. Esto protege sus CodeCatalyst recursos porque no puede eliminar inadvertidamente el permiso de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Sí en la columna Roles vinculados a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de rol vinculados al servicio para CodeCatalyst

CodeCatalyst usa el rol vinculado al servicio denominado `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`— Permite a Amazon acceder de CodeCatalyst solo lectura a los perfiles de las instancias de la aplicación y a los usuarios y grupos del directorio asociado en su nombre.

El rol vinculado al servicio

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` depende de los siguientes servicios para asumir el rol:

- `codecatalyst.amazonaws.com`

La política de permisos de roles denominada

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` permite CodeCatalyst realizar las siguientes acciones en los recursos especificados:

- Acción: `View application instance profiles and associated directory users and groups` para CodeCatalyst spaces that support identity federation and SSO users and groups

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para CodeCatalyst

No necesita crear manualmente un rol vinculado a servicios. Al crear un espacio en la AWS Management Console, la o la AWS API o la AWS CLI, se crea automáticamente el rol vinculado al servicio.

Important

Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Además, si utilizabas el CodeCatalyst servicio antes del 17 de noviembre de 2023, cuando comenzó a admitir roles vinculados al servicio, entonces CodeCatalyst creó el `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` rol en tu cuenta. Para obtener más información, consulte [Un nuevo rol ha aparecido en mi Cuenta de AWS](#).

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando creas un espacio, vuelve a CodeCatalyst para crear el rol vinculado al servicio para ti.

También puede usar la consola de IAM para crear un rol vinculado a un servicio con el caso de uso de View, perfiles de instancias de aplicaciones y directorios asociados, usuarios y grupos. En la AWS CLI o la API de AWS, cree un rol vinculado al servicio con el nombre de servicio `codecatalyst.amazonaws.com`. Para obtener más información, consulte [Creación de un rol vinculado a un servicio](#) en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Edición de un rol vinculado a un servicio para CodeCatalyst


CodeCatalyst no permite editar el rol vinculado al `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` servicio. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para CodeCatalyst

No es necesario eliminar manualmente el rol de `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Al eliminar un

espacio en la AWS Management Console, la API o la AWS CLI, se CodeCatalyst limpian los recursos y se elimina el rol vinculado al servicio.

También puede utilizar la consola de IAM, la AWS CLI o la API de AWS para eliminar manualmente el rol vinculado al servicio. Para ello, primero debe limpiar manualmente los recursos del rol vinculado al servicio para poder eliminarlo después manualmente.

 Note

Si el CodeCatalyst servicio utiliza el rol cuando intentas eliminar los recursos, es posible que la eliminación no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar CodeCatalyst los recursos utilizados por el `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`

- [Elimine el espacio.](#)

Cómo eliminar manualmente el rol vinculado a servicios mediante IAM

Puede usar la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a un servicio de `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Para obtener más información, consulte [Eliminar un rol vinculado a un servicio](#) en la Guía del usuario de IAM.

Regiones compatibles para funciones CodeCatalyst vinculadas al servicio

CodeCatalyst admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [Puntos de conexión y Regiones de AWS](#).

CodeCatalyst no admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Puede usar el rol `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` en las siguientes regiones.

Nombre de la región	Identidad de la región	Support en CodeCatalyst
Este de EE. UU. (Norte de Virginia)	us-east-1	No
Este de EE. UU. (Ohio)	us-east-2	No
Oeste de EE. UU. (Norte de California)	us-west-1	No
Oeste de EE. UU. (Oregón)	us-west-2	Sí
África (Ciudad del Cabo)	af-south-1	No
Asia-Pacífico (Hong Kong)	ap-east-1	No
Asia-Pacífico (Yakarta)	ap-southeast-3	No
Asia-Pacífico (Bombay)	ap-south-1	No
Asia Pacífico (Osaka)	ap-northeast-3	No
Asia Pacífico (Seúl)	ap-northeast-2	No
Asia Pacífico (Singapur)	ap-southeast-1	No
Asia Pacífico (Sídney)	ap-southeast-2	No
Asia Pacífico (Tokio)	ap-northeast-1	No
Canadá (centro)	ca-central-1	No
Europa (Fráncfort)	eu-central-1	No
Europa (Irlanda)	eu-west-1	Sí
Europa (Londres)	eu-west-2	No
Europa (Milán)	eu-south-1	No
Europa (París)	eu-west-3	No
Europa (Estocolmo)	eu-north-1	No

Nombre de la región	Identidad de la región	Support en CodeCatalyst
Medio Oriente (Baréin)	me-south-1	No
Medio Oriente (EAU)	me-central-1	No
América del Sur (São Paulo)	sa-east-1	No
AWS GovCloud (Este de EE. UU.)	us-gov-east-1	No
AWS GovCloud (EE. UU.-Oeste)	us-gov-west-1	No

AWSpolíticas gestionadas para Amazon CodeCatalyst

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas por AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir [políticas administradas por el cliente](#) específicas para los casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte [Políticas administradas por AWS](#) en la Guía del usuario de IAM.

Política administrada de AWS: AmazonCodeCatalystSupportAccess

Se trata de una política que concede permisos a todos los administradores y miembros del espacio para utilizar el plan de asistencia premium empresarial o empresarial asociado a la cuenta de facturación del espacio. Estos permisos permiten a los administradores y miembros del espacio utilizar el plan de soporte premium para los recursos para los que tienen permisos según las políticas de CodeCatalyst permisos.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- **support**— Otorga permisos para permitir a los usuarios buscar, crear y resolver casos de AWS Support. También otorga permisos para describir las comunicaciones, los niveles de gravedad, los archivos adjuntos y los detalles de los casos de soporte relacionados.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "support:DescribeAttachment",
        "support:DescribeCaseAttributes",
        "support:DescribeCases",
        "support:DescribeCommunications",
        "support:DescribeIssueTypes",
        "support:DescribeServices",
        "support:DescribeSeverityLevels",
        "support:DescribeSupportLevel",
        "support:SearchForCases",
        "support:AddAttachmentsToSet",
        "support:AddCommunicationToCase",
        "support:CreateCase",
        "support:InitiateCallForCase",
        "support:InitiateChatForCase",
        "support:PutCaseAttributes",
        "support:RateCaseCommunication",
        "support:ResolveCase"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }
]
}

```

Política administrada de AWS: AmazonCodeCatalystFullAccess

Se trata de una política que concede permisos para gestionar tu CodeCatalyst espacio y tus cuentas conectadas en la página Amazon CodeCatalyst Spaces delAWS Management Console. Esta aplicación se utiliza para configurar los espacios en los Cuentas de AWS que están conectados CodeCatalyst.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- `codecatalyst`— Otorga permisos completos para la página Amazon CodeCatalyst Spaces enAWS Management Console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeCatalystResourceAccess"
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCatalystAssociateIAMRole"
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"

```



```

    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "codecatalyst.amazonaws.com",
          "codecatalyst-runner.amazonaws.com"
        ]
      }
    }
  }
]
}

```

Política administrada de AWS: AmazonCodeCatalystReadOnlyAccess

Se trata de una política que concede permisos para ver y enumerar la información de los espacios y las cuentas conectadas en la página Amazon CodeCatalyst Spaces del AWS Management Console. Esta aplicación se utiliza para configurar Cuentas de AWS cuáles están conectados a su espacio en CodeCatalyst.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- `codecatalyst`— Otorga permisos de solo lectura a la página Amazon CodeCatalyst Spaces en AWS Management Console

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:Get*",
        "codecatalyst:List*"
      ],
      "Resource": "*"
    }
  ]
}

```

```

    }
  ]
}
```

Política administrada de AWS:

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy

No puede

adjuntar AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy; a sus entidades de IAM. Esta política está asociada a un rol vinculado al servicio que le permite CodeCatalyst realizar acciones en su nombre. Para obtener más información, consulte [Uso de roles vinculados a servicios de CodeCatalyst](#).

Esta política permite a los clientes ver los perfiles de las instancias de la aplicación y los usuarios y grupos del directorio asociado al administrar los espacios en ellos. CodeCatalyst Los clientes verán estos recursos cuando administren espacios que admitan la federación de identidades y los usuarios y grupos de SSO.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- sso— Otorga permisos que permiten a los usuarios ver los perfiles de instancias de aplicaciones que se administran en el Centro de Identidad de IAM para los espacios asociados. CodeCatalyst

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:ListApplications",
        "sso:ListApplicationAssignments",
        "sso:DescribeInstance",
```

```

    "sso:DescribeApplication"
  ],
  "Resource": "*"
}
]
}

```

Actualizaciones de CodeCatalyst en las políticas administradas de AWS

Consulte los detalles sobre las actualizaciones de las políticas AWS gestionadas CodeCatalyst desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del [historial del CodeCatalyst documento](#).

Cambio	Descripción	Fecha
AmazonCodeCatalyst ServiceRoleForIdentityCenterApplicationSynchronizationPolicy : política nueva	CodeCatalyst agregó la política. Otorga permisos que permiten a CodeCatalyst los usuarios ver los perfiles de las instancias de la aplicación y los usuarios y grupos del directorio asociado.	17 de noviembre de 2023
AmazonCodeCatalyst SupportAccess : política nueva	CodeCatalyst agregó la política. Otorga permisos que permiten a CodeCatalyst los usuarios buscar, crear y resolver casos de asistencia, así como ver las comunicaciones y los detalles relacionados.	20 de abril de 2023
AmazonCodeCatalyst FullAccess : política nueva	CodeCatalyst agregó la política.	20 de abril de 2023

Cambio	Descripción	Fecha
	Otorga acceso completo a CodeCatalyst.	
AmazonCodeCatalyst ReadOnlyAccess : política nueva	CodeCatalyst agregó la política. Otorga acceso de solo lectura a CodeCatalyst	20 de abril de 2023
CodeCatalyst comenzó a rastrear los cambios	CodeCatalyst comenzó a rastrear los cambios de sus políticas AWS gestionadas.	20 de abril de 2023

Funciones de IAM para el CodeCatalyst acceso de Amazon a los recursos AWS

CodeCatalyst puede acceder a AWS los recursos conectándolo Cuenta de AWS a un CodeCatalyst espacio. A continuación, puede crear los siguientes roles de servicio y asociarlos al conectar su cuenta.

Para obtener más información sobre los elementos que se utilizan en una política de JSON, consulte la [Referencia sobre los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM.

- Para acceder a los recursos en y Cuenta de AWS para tus CodeCatalyst proyectos y flujos de trabajo, primero debes conceder el permiso para acceder CodeCatalyst a esos recursos en tu nombre. Para ello, debe crear un rol de servicio en un entorno conectado Cuenta de AWS que CodeCatalyst pueda asumir en nombre de los usuarios y proyectos del espacio. Puede elegir entre crear y usar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName*servicio o puede crear roles de servicio personalizados y configurar estas políticas y roles de IAM manualmente. Como práctica recomendada, asigne a estas funciones la cantidad mínima de permisos necesaria.


Note

Para las funciones de servicio personalizadas, se requiere el director del CodeCatalyst servicio. Para obtener más información sobre el principio CodeCatalyst de servicio y el modelo de confianza, consulte [CodeCatalyst modelo de confianza](#).

- Para gestionar el soporte de un espacio a través del conectadoCuenta de AWS, puede optar por crear y utilizar el rol de `AWSRoleForCodeCatalystSupportservicio` que permite a CodeCatalyst los usuarios acceder al soporte. Para obtener más información sobre la compatibilidad con un CodeCatalyst espacio, consulte [AWS Supportpara Amazon CodeCatalyst](#).

Comprensión de la función `CodeCatalystWorkflowDevelopmentRole-spaceName` de servicio

Puede añadir un rol de IAM a su espacio que CodeCatalyst podrá utilizar para crear recursos y acceder a ellos en un entorno conectadoCuenta de AWS. Esto se denomina [rol de servicio](#). La forma más sencilla de crear un rol de servicio es agregar uno al crear el espacio y elegir la `CodeCatalystWorkflowDevelopmentRole-spaceName` opción para ese rol. Esto no solo crea el rol de servicio con el `AdministratorAccess` adjunto, sino que también crea la política de confianza que CodeCatalyst permite asumir el rol en nombre de los usuarios en los proyectos del espacio. La función de servicio se limita al espacio, no a proyectos individuales. Para crear esta función, consulte [Crear el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol para su cuenta y su espacio](#). Solo puedes crear un rol para cada espacio de cada cuenta.

 Note

Este rol solo se recomienda para su uso con cuentas de desarrollo y utiliza la política `AdministratorAccess` AWS administrada, lo que le da acceso total para crear nuevas políticas y recursos en ellaCuenta de AWS.

La política asociada a la `CodeCatalystWorkflowDevelopmentRole-spaceName` función está diseñada para funcionar con proyectos creados con planos en el espacio. Permite a los usuarios de esos proyectos desarrollar, crear, probar e implementar código utilizando los recursos de la red. Cuenta de AWS Para obtener más información, consulte [Crear un rol para un AWS servicio](#).

La política asociada a la `CodeCatalystWorkflowDevelopmentRole-spaceName` función es la política `AdministratorAccess` gestionada enAWS. Se trata de una política que otorga acceso total a todas AWS las acciones y recursos. Para ver el documento de política de JSON en la consola de IAM, consulte [AdministratorAccess](#).

La siguiente política de confianza permite CodeCatalyst asumir el `CodeCatalystWorkflowDevelopmentRole-spaceName` rol. Para obtener más información sobre el modelo de CodeCatalyst confianza, consulte [CodeCatalyst modelo de confianza](#).

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/*"
        }
      }
    }
  ]
}
```

Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol para su cuenta y su espacio

Sigue estos pasos para crear el CodeCatalystWorkflowDevelopmentRole-*spaceName* rol que se usará en los flujos de trabajo de tu espacio. Para cada cuenta en la que desees tener funciones de IAM para utilizarlas en proyectos, debes añadir a tu espacio una función, como la de desarrollador.

Antes de empezar, debe disponer de privilegios administrativos para su administrador Cuenta de AWS o poder trabajar con él. Para obtener más información sobre cómo Cuentas de AWS se utilizan las funciones de IAM CodeCatalyst, consulte [Cuentas de AWS Administrar un espacio](#).

Para crear y añadir el CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Antes de empezar en la CodeCatalyst consola, abra el espacio yAWS Management Console, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
3. Navega hasta tu CodeCatalyst espacio. Elija Settings (Configuración) y después Cuentas de AWS:

4. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de AWSdetalles.
5. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM al CodeCatalyst espacio de Amazon se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Es posible que deba iniciar sesión para acceder a la página.

6. Elija Crear función CodeCatalyst de administrador de desarrollo en IAM. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza del rol de desarrollo. El rol tendrá un nombreCodeCatalystWorkflowDevelopmentRole-*spaceName*. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName*de servicio](#).

Note

Este rol solo se recomienda para su uso con cuentas de desarrollador y usa la política AdministratorAccess AWS administrada, lo que le da acceso total para crear nuevas políticas y recursos en ellaCuenta de AWS.

7. Selecciona Crear rol de desarrollo.
8. En la página de conexiones, en la sección Funciones de IAM disponibles para CodeCatalyst, consulta la CodeCatalystWorkflowDevelopmentRole-*spaceName* función en la lista de funciones de IAM añadidas a tu cuenta.
9. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Entender la función AWSRoleForCodeCatalystSupportde servicio

Puede añadir una función de IAM a su espacio que CodeCatalyst los usuarios de un espacio puedan utilizar para crear casos de soporte y acceder a ellos. Esto se denomina [rol de servicio](#) de soporte. La forma más sencilla de crear un rol de servicio de soporte es añadir uno al crear el espacio y elegir la AWSRoleForCodeCatalystSupport opción para ese rol. Esto no solo crea la política y el rol, sino que también crea la política de confianza que permite CodeCatalyst asumir el rol en nombre de los usuarios en los proyectos del espacio. La función de servicio se limita al espacio, no a proyectos individuales. Para crear esta función, consulte [Crear el AWSRoleForCodeCatalystSupportrol para tu cuenta y tu espacio](#).

La política asociada a la `AWSRoleForCodeCatalystSupport` función es una política administrada que proporciona acceso a los permisos de soporte. Para obtener más información, consulte [Política administrada de AWS: AmazonCodeCatalystSupportAccess](#).

El rol de confianza de la política CodeCatalyst permite asumir el rol.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst.amazonaws.com",
          "codecatalyst-runner.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Crear el `AWSRoleForCodeCatalystSupport` rol para tu cuenta y tu espacio

Sigue estos pasos para crear el `AWSRoleForCodeCatalystSupport` rol que se usará para los casos de soporte en tu espacio. El rol debe agregarse a la cuenta de facturación designada para el espacio.

Antes de empezar, debe disponer de privilegios administrativos para su administrador Cuenta de AWS o poder trabajar con él. Para obtener más información sobre cómo Cuentas de AWS se utilizan las funciones de IAM CodeCatalyst, consulte [Cuentas de AWS Administrar un espacio](#).

Para crear y añadir el CodeCatalyst `AWSRoleForCodeCatalystSupport`

1. Antes de empezar en la CodeCatalyst consola, abra el espacio y `AWS Management Console`, a continuación, asegúrese de haber iniciado sesión con lo mismo Cuenta de AWS para su espacio.
2. Navega hasta tu CodeCatalyst espacio. Elija `Settings (Configuración)` y después `Cuentas de AWS`:
3. Elija el enlace en el Cuenta de AWS que desee crear el rol. Aparece la página de Cuenta de `AWS detalles`.

4. Elija Administrar roles desde AWS Management Console.

La página Añadir función de IAM al CodeCatalyst espacio de Amazon se abre en. AWS Management Console Esta es la página de Amazon CodeCatalyst Spaces. Puede que tengas que iniciar sesión para acceder a la página.

5. En los detalles del CodeCatalyst espacio, selecciona Add CodeCatalyst Support role. Esta opción crea un rol de servicio que contiene la política de permisos y la política de confianza para el rol de desarrollo preliminar. El rol tendrá un nombre `AWSRoleForCodeCatalystSupport` con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Entender la función AWSRoleForCodeCatalystSupport de servicio](#).
6. En la página Añadir función para CodeCatalyst Support, deje seleccionada la opción predeterminada y, a continuación, elija Crear función.
7. En Funciones de IAM disponibles para CodeCatalyst, consulta la `CodeCatalystWorkflowDevelopmentRole-spaceName` función en la lista de funciones de IAM añadidas a tu cuenta.
8. Para volver a tu espacio, selecciona Ir a Amazon CodeCatalyst.

Configurar las funciones de IAM para las acciones del flujo de trabajo en CodeCatalyst

En esta sección se detallan las funciones y políticas de IAM que puede crear para utilizarlas con su CodeCatalyst cuenta. Para obtener instrucciones sobre cómo crear roles de ejemplo, consulte [Crear roles manualmente para las acciones del flujo de trabajo](#). Tras crear el rol de IAM, copia el ARN del rol para añadir el rol de IAM a la conexión de tu cuenta y asociarlo al entorno de tu proyecto. Para obtener más información, consulte [Añadir funciones de IAM a las conexiones de cuentas](#).

CodeCatalyst rol de creación para el acceso a Amazon S3

Para las acciones de creación de CodeCatalyst flujos de trabajo, puede usar el rol de `CodeCatalystWorkflowDevelopmentRole-spaceName` servicio predeterminado o puede crear un rol de IAM denominado `CodeCatalystBuildRoleforS3Access`. Este rol usa una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en AWS CloudFormation sus recursos. Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- Escribe en los buckets de Amazon S3.
- Support the building of resources with AWS CloudFormation. Esto requiere acceso a Amazon S3.

Este rol usa la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  }]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

CodeCatalyst crear un rol para AWS CloudFormation

Para las acciones de creación de CodeCatalyst flujos de trabajo, puedes usar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminado o puedes crear un rol de IAM con los permisos necesarios. Este rol usa una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en sus AWS CloudFormation recursos. Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- Support the building of resources with AWS CloudFormation. Esto es obligatorio junto con la función de CodeCatalyst compilación para el acceso a Amazon S3 y la función de CodeCatalyst implementación para AWS CloudFormation.

Se deben adjuntar a esta función las siguientes políticas AWS administradas:

- AWSCloudFormationFullAccess

- IAM FullAccess
- Amazon S3 FullAccess
- API de Amazon GatewayAdministrator
- AWSLambdaFullAccess

CodeCatalyst rol de creación para CDK

Para CodeCatalyst los flujos de trabajo que ejecutan acciones de creación de CDK, como una aplicación web moderna de tres niveles, puede utilizar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminado o puede crear un rol de IAM con los permisos necesarios. Este rol usa una política con permisos específicos que CodeCatalyst necesita iniciar y ejecutar comandos de compilación de CDK para los recursos de su empresa. AWS CloudFormation Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- Escribe en los buckets de Amazon S3.
- Support la creación de construcciones de CDK y pilas de AWS CloudFormation recursos. Esto requiere acceso a Amazon S3 para el almacenamiento de artefactos, a Amazon ECR para el soporte de repositorios de imágenes y a SSM para el control y la supervisión del sistema para las instancias virtuales.

Este rol usa la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

CodeCatalyst implementar un rol para AWS CloudFormation

Para las acciones de implementación del CodeCatalyst flujo de trabajo que utilizan AWS CloudFormation, puede usar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminado o puede usar una política con permisos específicos que CodeCatalyst necesite ejecutar tareas en AWS CloudFormation los recursos de su Cuenta de AWS propiedad.

Este rol otorga permisos para hacer lo siguiente:

- Permite CodeCatalyst invocar una función λ para realizar un despliegue azul/verde de forma continua. AWS CloudFormation
- CodeCatalyst Permiten crear y actualizar pilas y conjuntos de cambios en. AWS CloudFormation

Este rol usa la siguiente política:

```

{"Action": [
  "cloudformation:CreateStack",
  "cloudformation>DeleteStack",
  "cloudformation:Describe*",
  "cloudformation:UpdateStack",
  "cloudformation:CreateChangeSet",
  "cloudformation>DeleteChangeSet",
  "cloudformation:ExecuteChangeSet",
  "cloudformation:SetStackPolicy",

```

```
    "cloudformation:ValidateTemplate",
    "cloudformation:List*",
    "iam:PassRole"
  ],
  "Resource": "resource_ARN",
  "Effect": "Allow"
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

CodeCatalyst rol de implementación para Amazon EC2

CodeCatalyst las acciones de despliegue del flujo de trabajo utilizan un rol de IAM con los permisos necesarios. Esta función utiliza una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en los recursos de Amazon EC2 de su empresa. Cuenta de AWS La política predeterminada del CodeCatalystWorkflowDevelopmentRole-*spaceName*rol no incluye los permisos para Amazon EC2 o Amazon EC2 Auto Scaling.

Este rol otorga permisos para hacer lo siguiente:

- Cree despliegues de Amazon EC2.
- Lea las etiquetas de una instancia o identifique una instancia de Amazon EC2 mediante los nombres de los grupos de Auto Scaling.
- Leer, crear, actualizar y eliminar grupos de Amazon EC2 Auto Scaling, enlaces de ciclo de vida y políticas de escalado.
- Publicar información para temas de Amazon SNS.
- Recupere información sobre CloudWatch las alarmas.
- Leer y actualizar Elastic Load Balancing.

Este rol usa la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling>DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling>CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:PutWarmPool",
        "autoscaling:DescribeScalingActivities",
        "autoscaling>DeleteAutoScalingGroup",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:TerminateInstances",
        "tag:GetResources",
        "sns:Publish",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets"
      ]
    }
  ]
}
```

```
],  
  "Resource": "resource_ARN"  
    }  
  ]  
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

CodeCatalyst rol de implementación para Amazon ECS

Para las acciones del CodeCatalyst flujo de trabajo, puede crear un rol de IAM con los permisos necesarios. Puede usar la función de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminada o puede crear una función de IAM para CodeCatalyst implementar acciones que se utilizarán en las implementaciones de Lambda. Esta función utiliza una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en los recursos de Amazon ECS de su Cuenta de AWS empresa.

Este rol otorga permisos para hacer lo siguiente:

- Inicie el despliegue continuo de Amazon ECS en nombre de un CodeCatalyst usuario, en una cuenta especificada en la CodeCatalyst conexión.
- Leer, actualizar y eliminar conjuntos de tareas de Amazon ECS.
- Actualizar grupos de destino de Elastic Load Balancing, oyentes y reglas.
- Invoque funciones Lambda.
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recuperar información sobre CloudWatch las alarmas.
- Publicar información para temas de Amazon SNS.

Este rol usa la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
      "codedeploy:ListApplications",
      "codedeploy:ListDeploymentGroups",
      "codedeploy:ListDeployments",
      "codedeploy:StopDeployment",
      "codedeploy:GetDeploymentTarget",
      "codedeploy:ListDeploymentTargets",
      "codedeploy:GetDeploymentConfig",
      "codedeploy:GetApplicationRevision",
      "codedeploy:RegisterApplicationRevision",
      "codedeploy:BatchGetApplicationRevisions",
      "codedeploy:BatchGetDeploymentGroups",
      "codedeploy:BatchGetDeployments",
      "codedeploy:BatchGetApplications",
      "codedeploy:ListApplicationRevisions",
```



```

    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
    "iam:PassRole"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

CodeCatalyst rol de implementación para Lambda

Para las acciones CodeCatalyst del flujo de trabajo, puede crear un rol de IAM con los permisos necesarios. Puede usar la función de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminada o crear una función de IAM para CodeCatalyst implementar acciones que se utilizarán en las implementaciones de Lambda. Esta función utiliza una política con permisos limitados que CodeCatalyst necesita para ejecutar tareas en los recursos de Lambda de su empresa.

Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- Lea, actualice e invoque funciones y alias de Lambda.

- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre CloudWatch las alarmas de eventos.
- Publicar información para temas de Amazon SNS.

Este rol usa la siguiente política:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/CodeDeploy/",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
      },
      "Effect": "Allow"
    },
    {
```

```

        "Action": [
            "lambda:InvokeFunction"
        ],
        "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
        "Effect": "Allow"
    }
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

CodeCatalyst rol de implementación para Lambda

Para las acciones del CodeCatalyst flujo de trabajo, puede usar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminado o puede crear un rol de IAM con los permisos necesarios. Esta función utiliza una política con permisos limitados que CodeCatalyst necesita para ejecutar tareas en los recursos de Lambda de su empresa. Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- Lea, actualice e invoque funciones y alias de Lambda.
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre las alarmas. CloudWatch
- Publicar información para temas de Amazon SNS.

Este rol usa la siguiente política:

```

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

```

```

    "Action": [
      "cloudwatch:DescribeAlarms",
      "lambda:UpdateAlias",
      "lambda:GetAlias",
      "lambda:GetProvisionedConcurrencyConfig",
      "sns:Publish"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:::function:CodeDeployHook_*",
    "Effect": "Allow"
  }
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

CodeCatalyst implementar un rol para AWS SAM

Para las acciones del CodeCatalyst flujo de trabajo, puede utilizar el rol de CodeCatalystWorkflowDevelopmentRole-*spaceName* servicio predeterminado o puede crear un rol de IAM con los permisos necesarios. Este rol usa una política con permisos específicos que CodeCatalyst necesita para ejecutar las tareas AWS SAM y AWS CloudFormation los recursos de su empresa. Cuenta de AWS

Este rol otorga permisos para hacer lo siguiente:

- CodeCatalyst Permiten invocar una función Lambda para realizar el despliegue de aplicaciones CLI AWS SAM y sin servidor.
- CodeCatalyst Permiten crear y actualizar pilas y conjuntos de cambios en. AWS CloudFormation

Este rol usa la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",

```

```

        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
    ],
    "Resource": "*"
}
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

CodeCatalyst función de solo lectura para Amazon EC2

Para las acciones CodeCatalyst del flujo de trabajo, puede crear un rol de IAM con los permisos necesarios. Esta función utiliza una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en los recursos de Amazon EC2 de su empresa. Cuenta de AWS La función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio no incluye los permisos para Amazon EC2 ni las acciones descritas para Amazon. CloudWatch

Este rol otorga permisos para hacer lo siguiente:

- Obtenga el estado de las instancias de Amazon EC2.
- Obtenga CloudWatch métricas para las instancias de Amazon EC2.

Este rol usa la siguiente política:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe",
      "Resource": "resource_ARN"
    }
  ]
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe"
      ],
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe",
      "Resource": "resource_ARN"
    }
  ]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

CodeCatalyst función de solo lectura para Amazon ECS

Para las acciones del CodeCatalyst flujo de trabajo, puede crear un rol de IAM con los permisos necesarios. Esta función utiliza una política con permisos específicos que CodeCatalyst necesita para ejecutar tareas en los recursos de Amazon ECS de su Cuenta de AWS empresa.

Este rol otorga permisos para hacer lo siguiente:

- Lea los conjuntos de tareas de Amazon ECS.

- Recupere información sobre CloudWatch las alarmas.

Este rol usa la siguiente política:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:DescribeServices",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeRules"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
      },
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": [
```



```

        "arn:aws:iam:::role/ecsTaskExecutionRole",
        "arn:aws:iam:::role/ECSTaskExecution"
    ],
    "Condition": {
        "StringLike": {
            "iam:PassedToService": [
                "ecs-tasks.amazonaws.com"
            ]
        }
    }
}
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

CodeCatalyst función de solo lectura para Lambda

Para las acciones CodeCatalyst del flujo de trabajo, puede crear un rol de IAM con los permisos necesarios. Esta función utiliza una política con permisos limitados que CodeCatalyst necesita para ejecutar tareas en los recursos de Lambda de su empresa. Cuenta de AWS

Este rol otorga permisos para lo siguiente:

- Lea las funciones y los alias de Lambda.
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre CloudWatch las alarmas.

Este rol utiliza la siguiente política de .

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

```

"Statement": [
  {
    "Action": [
      "cloudwatch:DescribeAlarms",
      "lambda:GetAlias",
      "lambda:GetProvisionedConcurrencyConfig"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  }
]
}

```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"

```

Crear roles manualmente para las acciones del flujo de trabajo

CodeCatalyst Las acciones del flujo de trabajo utilizan las funciones de IAM que usted cree, denominadas función de creación, función de implementación y función de pila.

Siga estos pasos para crear estos roles en IAM.

Para crear un rol de despliegue

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Elija Create Policy (Crear política).
 - e. Seleccione la pestaña JSON.
 - f. Elimine el código existente.
 - g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-deploy-policy
```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

- 2. Cree el rol de implementación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    ]  
  }
```

- e. Elija Siguiente.
- f. En Políticas de permisos, busque `codecatalyst-deploy-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

```
codecatalyst-deploy-role
```

- i. En la descripción del rol, introduzca:

```
CodeCatalyst deploy role
```

- j. Elija Crear rol.

Ahora ha creado un rol de despliegue con una política de confianza y una política de permisos.

3. Obtenga el ARN del rol de implementación de la siguiente manera:

- a. Seleccione Roles en el panel de navegación.
- b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-deploy-role`).
- c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de implementación con los permisos adecuados y ha obtenido su ARN.

Para crear un rol de compilación

1. Cree una política para el rol, de la siguiente manera:

- a. Inicie sesión en AWS.
- b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

- d. Elija Create Policy (Crear política).
- e. Seleccione la pestaña JSON.
- f. Elimine el código existente.
- g. Pegue el siguiente código:

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

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

```
codecatalyst-build-policy
```

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:

- a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
- b. Elija Política de confianza personalizada.

- c. Elimine la política de confianza personalizada existente.
- d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. En Políticas de permisos, busque `codecatalyst-build-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-build-role

- i. En la descripción del rol, introduzca:

CodeCatalyst build role

- j. Elija Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

3. Obtenga el ARN del rol de compilación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.


- b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (codecatalyst-build-role).
- c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de compilación con los permisos adecuados y ha obtenido su ARN.

Para crear un rol de pila

 Note

No es necesario crear un rol de pila, aunque se recomienda hacerlo por motivos de seguridad. Si no crea la función de pila, tendrá que añadir las políticas de permisos que se describen más adelante en este procedimiento a la función de implementación.

1. Inicia sesión AWS con la cuenta en la que quieres implementar tu pila.
2. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
3. En el panel de navegación, selecciona Funciones y, a continuación, selecciona Crear función.
4. En la parte superior, selecciona AWSservicio.
5. De la lista de servicios, selecciona CloudFormation.
6. Elija Siguiente: permisos.
7. En el cuadro de búsqueda, agrega las políticas necesarias para acceder a los recursos de tu pila. Por ejemplo, si su pila incluye una AWS Lambda función, debe añadir una política que conceda acceso a Lambda.

 Tip

Si no está seguro de qué políticas añadir, puede omitirlas por ahora. Cuando pruebas la acción, si no tienes los permisos adecuados, AWS CloudFormation genera errores que muestran qué permisos debes añadir.

8. Elija Siguiente: Etiquetas.

9. Elija Siguiente: Revisar.
10. En Nombre del rol, ingresa:

`codecatalyst-stack-role`

11. Elija Crear rol.
12. Para obtener el ARN del rol de pila, haga lo siguiente:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-stack-role`).
 - c. Elija el rol de la lista.
 - d. En la página de resumen, copie el valor del ARN del rol.

Se utiliza AWS CloudFormation para crear políticas y funciones en IAM

Puede optar por crear y utilizar AWS CloudFormation plantillas para crear las políticas y funciones que necesita para acceder a los recursos y Cuenta de AWS para sus CodeCatalyst proyectos y flujos de trabajo. AWS CloudFormation es un servicio que le ayuda a modelar y configurar sus AWS recursos para que pueda dedicar menos tiempo a gestionarlos y más tiempo a centrarse en las aplicaciones en las que se ejecutan AWS. Si tiene la intención de crear funciones en varias Cuentas de AWS, la creación de una plantilla puede ayudarle a realizar esta tarea con mayor rapidez.

La siguiente plantilla de ejemplo crea una política y un rol de acción de despliegue.

```
Parameters:
  CodeCatalystAccountId:
    Type: String
    Description: Account ID from the connections page
  ExternalId:
    Type: String
    Description: External ID from the connections page
Resources:
  CrossAccountRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
```

```
- Effect: Allow
Principal:
  AWS:
    - !Ref CodeCatalystAccountId
Action:
  - 'sts:AssumeRole'
Condition:
  StringEquals:
    sts:ExternalId: !Ref ExternalId
Path: /
Policies:
  - PolicyName: CodeCatalyst-CloudFormation-action-policy
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Action:
            - 'cloudformation:CreateStack'
            - 'cloudformation>DeleteStack'
            - 'cloudformation:Describe*'
            - 'cloudformation:UpdateStack'
            - 'cloudformation:CreateChangeSet'
            - 'cloudformation>DeleteChangeSet'
            - 'cloudformation:ExecuteChangeSet'
            - 'cloudformation:SetStackPolicy'
            - 'cloudformation:ValidateTemplate'
            - 'cloudformation:List*'
            - 'iam:PassRole'
          Resource: '*'
```

Crear el rol manualmente para el esquema de la aplicación web

El esquema de la aplicación CodeCatalyst web utiliza las funciones de IAM que usted crea, denominadas función de creación para CDK, función de implementación y función de pila.

Siga estos pasos para crear el rol en IAM.

Para crear un rol de compilación

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

- c. En el panel de navegación, seleccione Políticas.
- d. Seleccione Crear política.
- e. Seleccione la pestaña JSON.
- f. Elimine el código existente.
- g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

codecatalyst-webapp-build-policy

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. Adjunta la política de permisos a la función de compilación. En la página Añadir permisos, en la sección Políticas de permisos, busque codecatalyst-webapp-build-policy y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-webapp-build-role

- i. En la descripción del rol, introduzca:

CodeCatalyst Web app build role

- j. Elija Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

3. Adjunta la política de permisos al rol de compilación de la siguiente manera:
 - a. En el panel de navegación, elija Roles y, a continuación, busque `codecatalyst-webapp-build-role`.
 - b. Elija `codecatalyst-webapp-build-role` mostrar sus detalles.
 - c. En la pestaña Permisos, selecciona Añadir permisos y, a continuación, selecciona Adjuntar políticas.
 - d. `Busquecodecatalyst-webapp-build-policy`, active su casilla de verificación y, a continuación, elija Adjuntar políticas.

Ahora ha adjuntado la política de permisos a la función de creación. La función de creación ahora tiene dos políticas: una política de permisos y una política de confianza.

4. Obtenga el ARN del rol de compilación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-webapp-build-role`).
 - c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de compilación con los permisos adecuados y ha obtenido su ARN.

Creación manual de funciones para el plan SAM

El plan CodeCatalyst SAM utiliza las funciones de IAM que usted crea, denominadas función de creación CloudFormation y función de implementación para SAM.

Siga estos pasos para crear los roles en IAM.

Para crear un rol de compilación para CloudFormation

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 - c. En el panel de navegación, seleccione Políticas.
 - d. Seleccione Crear política.
 - e. Seleccione la pestaña JSON.
 - f. Elimine el código existente.
 - g. Pegue el siguiente código:

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

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.
- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

codecatalyst-SAM-build-policy

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:
 - a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
 - b. Elija Política de confianza personalizada.
 - c. Elimine la política de confianza personalizada existente.
 - d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. Adjunta la política de permisos a la función de compilación. En la página Añadir permisos, en la sección Políticas de permisos, busque codecatalyst-SAM-build-policy y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

codecatalyst-SAM-build-role

- i. En la descripción del rol, introduzca:

CodeCatalyst SAM build role

- j. Elija Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

3. Adjunta la política de permisos al rol de compilación de la siguiente manera:
 - a. En el panel de navegación, elija Roles y, a continuación, busque `codecatalyst-SAM-build-role`.
 - b. Elija `codecatalyst-SAM-build-role` mostrar sus detalles.
 - c. En la pestaña Permisos, selecciona Añadir permisos y, a continuación, selecciona Adjuntar políticas.
 - d. Busque `codecatalyst-SAM-build-policy`, active su casilla de verificación y, a continuación, elija Adjuntar políticas.

Ahora ha adjuntado la política de permisos a la función de creación. La función de creación ahora tiene dos políticas: una política de permisos y una política de confianza.

4. Obtenga el ARN del rol de compilación de la siguiente manera:
 - a. Seleccione Roles en el panel de navegación.
 - b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-SAM-build-role`).
 - c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de compilación con los permisos adecuados y ha obtenido su ARN.

Para crear un rol de implementación para SAM

1. Cree una política para el rol, de la siguiente manera:
 - a. Inicie sesión en AWS.
 - b. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

- c. En el panel de navegación, seleccione Políticas.
- d. Seleccione Crear política.
- e. Seleccione la pestaña JSON.
- f. Elimine el código existente.
- g. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La primera vez que se utilice el rol para ejecutar acciones de flujo de trabajo, utilice el comodín en la declaración de política de recursos y, a continuación, defina la política con el nombre del recurso cuando esté disponible.

```
"Resource": "*"
```

- h. Elija Siguiente: Etiquetas.

- i. Elija Siguiente: Revisar.
- j. En Nombre, introduzca:

`codecatalyst-SAM-deploy-policy`

- k. Elija Crear política.

Ahora ha creado una política de permisos.

2. Cree el rol de compilación de la siguiente manera:

- a. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
- b. Elija Política de confianza personalizada.
- c. Elimine la política de confianza personalizada existente.
- d. Añada la siguiente política de confianza personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Elija Siguiente.
- f. Adjunta la política de permisos a la función de compilación. En la página Añadir permisos, en la sección Políticas de permisos, busque `codecatalyst-SAM-deploy-policy` y active su casilla de verificación.
- g. Elija Siguiente.
- h. En Nombre del rol, escriba:

`codecatalyst-SAM-deploy-role`

- i. En la descripción del rol, introduzca:

`CodeCatalyst SAM deploy role`

- j. Elija Crear rol.

Ahora ha creado un rol de compilación con una política de confianza y una política de permisos.

3. Adjunta la política de permisos al rol de compilación de la siguiente manera:

- a. En el panel de navegación, elija Roles y, a continuación, busque `codecatalyst-SAM-deploy-role`.
- b. Elija `codecatalyst-SAM-deploy-role` mostrar sus detalles.
- c. En la pestaña Permisos, selecciona Añadir permisos y, a continuación, selecciona Adjuntar políticas.
- d. Busque `codecatalyst-SAM-deploy-policy`, active su casilla de verificación y, a continuación, elija Adjuntar políticas.

Ahora ha adjuntado la política de permisos a la función de creación. La función de creación ahora tiene dos políticas: una política de permisos y una política de confianza.

4. Obtenga el ARN del rol de compilación de la siguiente manera:

- a. Seleccione Roles en el panel de navegación.
- b. En el cuadro de búsqueda, introduce el nombre del rol que acabas de crear (`codecatalyst-SAM-deploy-role`).
- c. Elija el rol de la lista.

Aparece la página de resumen del rol.

- d. En la parte superior, copia el valor del ARN.

Ahora ha creado el rol de compilación con los permisos adecuados y ha obtenido su ARN.

Validación de conformidad para Amazon CodeCatalyst

Para saber si un Servicio de AWS está incluido en el ámbito de programas de conformidad específicos, consulte [Servicios de AWS en el ámbito del programa de conformidad](#) y escoja el programa de conformidad que le interese. Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la sensibilidad de los datos, los destinos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- [Guías de inicio rápido de seguridad y conformidad](#): estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) (Arquitectura para la seguridad y el cumplimiento de la HIPAA en Amazon Web Services): en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptas para HIPAA.

Note

No todos los Servicios de AWS son aptos para HIPAA. Para obtener más información, consulte la [Referencia de servicios aptos para HIPAA](#).

- [Recursos de conformidad de AWS](#): este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [AWS Guías de cumplimiento para clientes](#): comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- [Evaluación de recursos con reglas](#) en la Guía para desarrolladores de AWS Config: el servicio AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.

- [AWS Security Hub](#): este Servicio de AWS proporciona una visión completa de su estado de seguridad en AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte [la referencia de controles de Security Hub](#).
- [AWS Audit Manager](#): este Servicio de AWS le ayuda a auditar continuamente el uso de AWS con el fin de simplificar la forma en que administra el riesgo y la conformidad con las normativas y los estándares del sector.

Resiliencia en Amazon CodeCatalyst

La infraestructura global de AWS se construye en torno a las Regiones de AWS y a las zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja demora. Mediante las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre las Regiones de AWS y las zonas de disponibilidad, consulte la [Infraestructura global de AWS](#). Para obtener más información sobre CodeCatalyst los datos en los que se replican Regiones de AWS, consulte [Protección de datos en Amazon CodeCatalyst](#).

Seguridad de infraestructuras en Amazon CodeCatalyst

Como servicio gestionado, Amazon CodeCatalyst está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte [Seguridad en la nube de AWS](#). Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte [Protección de la infraestructura](#) en Portal de seguridad de AWS Well-Architected Framework.

Utilizas las llamadas a la API AWS publicadas para acceder CodeCatalyst a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Análisis de configuración y vulnerabilidad en Amazon CodeCatalyst

La configuración y los controles de TI son una responsabilidad compartida entre AWS y usted, nuestro cliente. Para obtener más información, consulte el [modelo de responsabilidad compartida de AWS](#).

Tus datos y privacidad en Amazon CodeCatalyst

Amazon CodeCatalyst se toma muy en serio tu privacidad y la seguridad de tu información es nuestra principal prioridad. Puedes obtener más información sobre cómo gestionamos tu información en el [Aviso AWS de privacidad](#).

Para solicitar y ver sus datos, consulte [Solicitar sus datos](#) en elReferencia general de AWS.

Eliminar tu perfil de AWS Builder ID

Eliminar tu perfil es una acción permanente que no se puede revertir. El proceso de eliminación comienza inmediatamente después de seleccionar Eliminar. Amazon CodeCatalyst comienza a eliminar tu perfil y toda la información personal asociada. Este proceso puede tardar hasta 90 días en completarse.

Cuando se elimina tu perfil, no podrás acceder a tus datos ni recuperarlos en Amazon CodeCatalyst. Esto incluye los tokens de acceso personal, los roles, las membresías de usuario y cualquier CodeCatalyst espacio de Amazon del que seas el único miembro. Ya no puedes iniciar sesión en Amazon CodeCatalyst.

Para obtener información sobre cómo eliminar tu perfil de AWS Builder ID, consulta [Eliminar tu AWS Builder ID](#) en elReferencia general de AWS.

Mejores prácticas para las acciones de flujo de trabajo en Amazon CodeCatalyst

Hay una serie de prácticas recomendadas de seguridad que se deben tener en cuenta a la hora de desarrollar los flujos de trabajo CodeCatalyst. Las siguientes son pautas generales y no representan una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no

sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

Temas

- [Información confidencial](#)
- [Términos de licencia](#)
- [Código que no es de confianza](#)
- [GitHub Acciones](#)

Información confidencial

No insertes información confidencial en tu YAML. En lugar de incrustar credenciales, claves o símbolos en tu YAML, te recomendamos que utilices secretos. CodeCatalyst Los secretos proporcionan una forma sencilla de almacenar y hacer referencia a información confidencial desde tu YAML.

Términos de licencia

Asegúrate de prestar atención a las condiciones de licencia de la acción que decidas utilizar.

Código que no es de confianza

Por lo general, las acciones son módulos autónomos de un solo propósito que se pueden compartir en un proyecto, un espacio o la comunidad en general. El uso de código de otros usuarios puede suponer una gran ventaja en términos de comodidad y eficiencia, pero también introduce un nuevo vector de amenazas. Revise las siguientes secciones para asegurarse de seguir las mejores prácticas para mantener seguros sus flujos de trabajo de CI/CD.

GitHub Acciones

GitHub Las acciones son de código abierto, creadas y mantenidas por la comunidad. Seguimos el [modelo de responsabilidad compartida](#) y consideramos el código fuente de GitHub Actions como datos de clientes de los que tú eres responsable. GitHub A las acciones se les puede conceder acceso a los secretos, a los tokens de los repositorios, al código fuente, a los enlaces de las cuentas y a tu tiempo de cómputo. Asegúrate de confiar en la fiabilidad y la seguridad de las GitHub acciones que planeas ejecutar.

Directrices más específicas y mejores prácticas de seguridad para Actions: GitHub

- [Fortalecimiento de la seguridad](#)
- [Prevenir las propias solicitudes](#)
- [Entrada que no es de confianza](#)
- [¿Cómo confiar en tus componentes](#)

CodeCatalyst modelo de confianza

El modelo de CodeCatalyst confianza de Amazon CodeCatalyst permite asumir el rol de servicio en lo conectadoCuenta de AWS. El modelo conecta la función de IAM, los principios del CodeCatalyst servicio y el CodeCatalyst espacio. La política de confianza utiliza la clave de `aws:SourceArn` condición para conceder permisos al CodeCatalyst espacio especificado en la clave de condición. Para obtener más información sobre esta clave de condición, consulte [aws: SourceArn](#) en la Guía del usuario de IAM.

Una política de confianza es un documento de política de JSON en el que se definen los principios en los que se confía para asumir la función. Una política de confianza de rol es una política basada en recursos requerida que se adjunta a un rol en IAM. Para obtener más información, consulte [los términos y conceptos](#) en la Guía del usuario de IAM. Para obtener más información sobre los principios del servicio CodeCatalyst, consulte. [Principios de servicio para CodeCatalyst](#)

En la siguiente política de confianza, los principales de servicio que figuran en el `Principal` elemento reciben permisos de la política basada en recursos y el `Condition` bloqueo se utiliza para limitar el acceso al recurso limitado.

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/*"
        }
      }
    }
  ]
```



```
    }  
  }  
]
```

En la política de confianza, los directores del CodeCatalyst servicio tienen acceso a través de la clave de `aws:SourceArn` condición, que contiene el nombre de recurso de Amazon (ARN) del ID CodeCatalyst del espacio. El ARN utiliza el siguiente formato:

```
arn:aws:codecatalyst:::space/spaceId/project/*
```

Important

Utilice el ID de espacio solo en las claves de condición, como `aws:SourceArn`. No utilice el ID de espacio en las declaraciones de política de IAM como un ARN de recurso.

Como práctica recomendada, limite los permisos tanto como sea posible en la política.

- Puede usar el comodín (*) en la clave de `aws:SourceArn` condición para especificar todos los proyectos del espacio con `project/*`.
- Puede especificar los permisos a nivel de recursos en la clave de `aws:SourceArn` condición para un proyecto específico en el espacio con `project/projectId`

Principios de servicio para CodeCatalyst

El `Principal` elemento se utiliza en una política de JSON basada en recursos para especificar el principal al que se le permite o deniega el acceso a un recurso. Las entidades principales que puede especificar en la política de confianza incluyen usuarios, roles, cuentas y servicios. No se puede utilizar el `Principal` elemento en una política basada en la identidad; del mismo modo, no se puede identificar a un grupo de usuarios como principal en una política (por ejemplo, una política basada en recursos) porque los grupos se refieren a los permisos, no a la autenticación, y los principales son entidades de IAM autenticadas.

En la política de confianza, puede especificar en el `Principal` elemento de una política basada en recursos o en condiciones las claves que admitan a los principales. Los principios del servicio los define el servicio. Los principios de servicio definidos para: CodeCatalyst

- `codecatalyst.amazonaws.com`: este principio de servicio se utiliza para un rol al que se concederá el acceso. CodeCatalyst AWS
- `codecatalyst-runner.amazonaws.com`: este principio de servicio se utiliza para un rol que permitirá el acceso a los recursos en las implementaciones de los flujos de trabajo. CodeCatalyst AWS CodeCatalyst

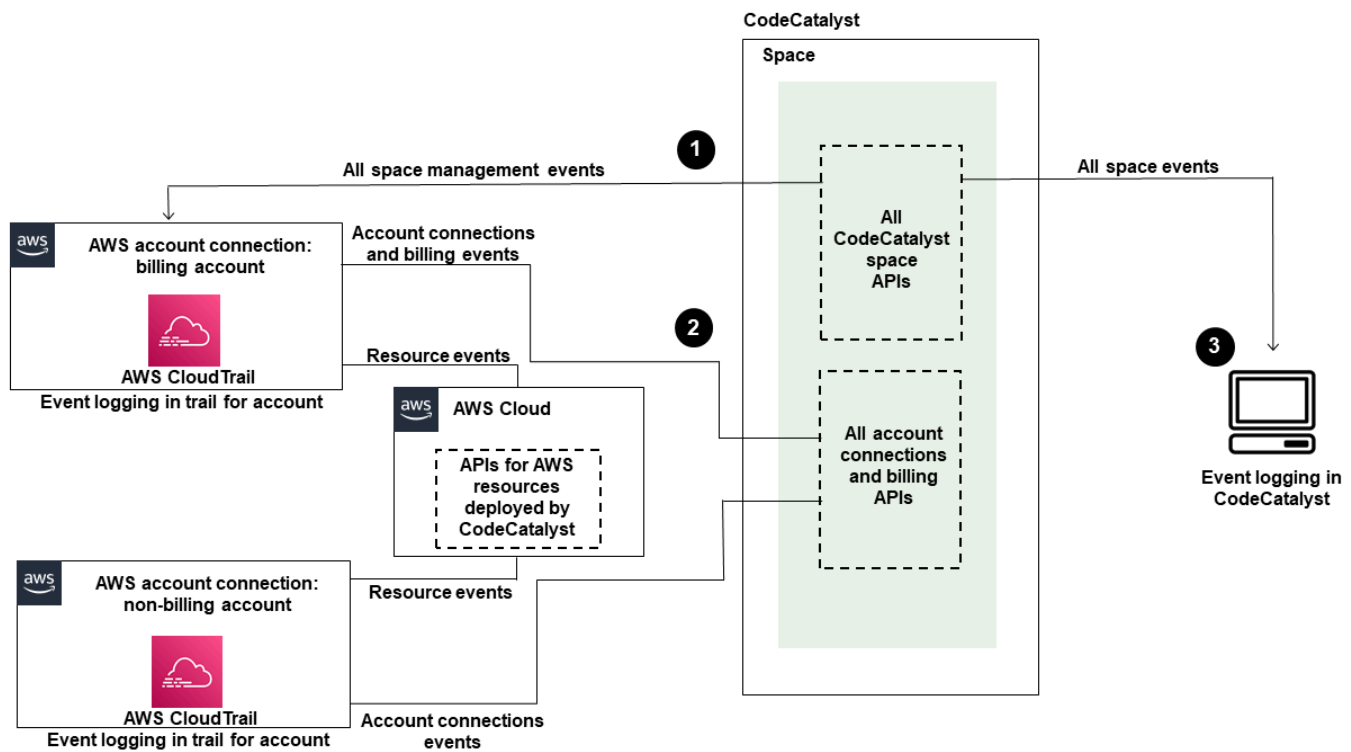
Para obtener más información, consulte Elementos principales de la [política de AWS JSON](#) en la Guía del usuario de IAM.

Monitorización en Amazon CodeCatalyst

En Amazon CodeCatalyst, los eventos de administración del espacio se recopilan AWS CloudTrail y se registran en el registro de la cuenta de facturación del espacio. CloudTrail El registro es el método principal para gestionar el registro de CodeCatalyst eventos, y un método secundario es ver el registro de eventos CodeCatalyst.

Los eventos de la cuenta se registran con la ruta y el depósito designado que están configurados para el Cuenta de AWS.


En el siguiente diagrama se muestra cómo se registran todos los eventos de administración del espacio en CloudTrail la cuenta de facturación, mientras que las conexiones o eventos de facturación y los eventos de AWS recursos se registran en la cuenta CloudTrail correspondiente.




El siguiente diagrama muestra los siguientes pasos:

1. Cuando se crea un espacio, una Cuenta de AWS se conecta al espacio y se designa como cuenta de facturación. El registro utilizado es el registro que se creó CloudTrail para la cuenta de facturación, donde se registran los eventos relacionados con el espacio. CloudTrail captura las llamadas a la API y los eventos relacionados realizados por un CodeCatalyst espacio o en su nombre y envía los archivos de registro a un depósito de S3 que usted especifique. Si la cuenta de facturación cambia a otra cuenta de AWS, los eventos espaciales se registran en el registro y el bucket de esa cuenta. Para obtener más información sobre los eventos de CodeCatalyst administración que registra CloudTrail, consulte [CodeCatalyst información en CloudTrail](#).
2. Otras cuentas conectadas al espacio, incluida la cuenta de facturación, registran un subconjunto de eventos para las conexiones de cuentas y los eventos de facturación. CodeCatalyst Los flujos de trabajo que generan eventos de cuenta para AWS los recursos desplegados en esa cuenta también se registran en el registro y en el Cuenta de AWS bucket de. CloudTrail captura las llamadas a la API y los eventos relacionados realizados por un CodeCatalyst espacio o en su nombre y envía los archivos de registro a un depósito de S3 que usted especifique. Para obtener más información sobre los eventos de CodeCatalyst administración que registra CloudTrail, consulte [Acceder a los eventos registrados en CodeCatalyst](#).

3. También puede supervisar CodeCatalyst las acciones en su espacio dentro de un tiempo específico en el espacio con el [list-event-logs](#) comando que utiliza el AWS CLI. Para obtener más información, consulta [la Guía de referencia de las CodeCatalyst API de Amazon](#). Debes tener la función de administrador del espacio para acceder a la lista de eventos y CodeCatalyst realizar acciones en tu espacio. Para obtener más información, consulte [Acceder a los eventos registrados en CodeCatalyst](#).

 Note

ListEventLogs garantiza los eventos de los últimos 30 días en un espacio determinado. También puede ver y recuperar una lista de los eventos de administración de los últimos 90 días CodeCatalyst en la AWS CloudTrail consola consultando el historial de eventos o creando un registro para crear y mantener un registro de los eventos que se prolonguen más de 90 días. Para obtener más información, consulte [Trabajar con el historial de CloudTrail eventos](#) y [Trabajar con CloudTrail rutas](#).

 Note

AWS los recursos que se implementan en las cuentas conectadas para los CodeCatalyst flujos de trabajo no se registran como parte del CloudTrail registro del CodeCatalyst espacio. Por ejemplo, CodeCatalyst los recursos incluyen un espacio o un proyecto. AWS los recursos incluyen un servicio Amazon ECS o una función Lambda. Debe configurar el CloudTrail registro por separado para cada Cuenta de AWS lugar en el que se implementen los recursos.

Este es un flujo posible para la supervisión de eventos CodeCatalyst.

Mary Major es la administradora de un CodeCatalyst espacio y consulta todos los eventos de gestión en CodeCatalyst busca de recursos a nivel de espacio y de proyecto en el espacio en el que se ha iniciado sesión. CloudTrail Consulte, [CodeCatalyst información en CloudTrail](#) por ejemplo, los eventos en los que se ha iniciado sesión. CloudTrail

En el caso de los recursos que se crean en CodeCatalyst, como los entornos de desarrollo, Mary consulta el historial de eventos en la cuenta de facturación del espacio e investiga los eventos en CodeCatalyst los que los miembros del proyecto crearon entornos de desarrollo. El evento proporciona el almacén de identidades, el tipo de identidad de IAM y las credenciales para el AWS

ID de creación del usuario que creó el entorno de desarrollo. En el caso de los recursos que se crean AWS cuando los flujos de trabajo los implementan CodeCatalyst, como una función Lambda para un despliegue sin servidor, el Cuenta de AWS propietario puede ver el historial de eventos del registro asociado a la acción independiente Cuenta de AWS (que también es una cuenta conectada CodeCatalyst) de la acción de despliegue del flujo de trabajo.

Para investigar más a fondo, Mary también puede ver los eventos de todas CodeCatalyst las API del espacio mediante el [list-event-logs](#) comando del. AWS CLI

Temas

- [Registro de llamadas a la CodeCatalyst API en Connected Cuentas de AWS mediante AWS CloudTrail](#)
- [Acceder a los eventos registrados en CodeCatalyst](#)

Registro de llamadas a la CodeCatalyst API en Connected Cuentas de AWS mediante AWS CloudTrail

Amazon CodeCatalyst está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura las llamadas a la API realizadas en nombre de CodeCatalyst in connected Cuentas de AWS as events. Si crea un registro, puede habilitar la entrega continua de CloudTrail eventos a un bucket de S3, incluidos los eventos de CodeCatalyst. Si no configuras una ruta, podrás ver los eventos más recientes en la CloudTrail consola, en el historial de eventos.

CodeCatalyst admite el registro de las siguientes acciones como eventos en los archivos de CloudTrail registro:

- Los eventos de administración de CodeCatalyst los espacios se registrarán en la Cuenta de AWS cuenta de facturación designada para el espacio. Para obtener más información, consulte [CodeCatalyst eventos espaciales](#).

Note

Se puede acceder a los eventos de datos de los CodeCatalyst espacios mediante la CLI, como se detalla en [Acceder a los eventos registrados en CodeCatalyst](#).


- Los eventos de los recursos que se utilizan en las acciones del CodeCatalyst flujo de trabajo que se producen en un Cuenta de AWS entorno conectado se registrarán como eventos en ese

entornoCuenta de AWS. Para obtener más información, consulte [CodeCatalyst conexiones de cuentas y eventos de facturación](#).

 Important

Si bien se pueden asociar varias cuentas a un espacio, el CloudTrail registro de eventos en CodeCatalyst espacios y proyectos solo se aplica a la cuenta de facturación.

La cuenta de facturación de espacios es tuya y se cobran los CodeCatalyst recursos Cuenta de AWS que superen el nivel AWS gratuito. Se pueden conectar varias cuentas a un espacio, mientras que solo una cuenta puede ser la cuenta de facturación designada. La cuenta de facturación o las cuentas conectadas adicionales del espacio pueden tener funciones de IAM que se utilizan para implementar AWS recursos e infraestructura, como un clúster de Amazon ECS o un bucket de S3, a partir de CodeCatalyst flujos de trabajo. Puede usar el flujo de trabajo YAML para identificar el flujo de trabajo en el Cuenta de AWS que realizó la implementación.

 Note

AWS los recursos que se implementan en las cuentas conectadas para los CodeCatalyst flujos de trabajo no se registran como parte del CloudTrail registro del CodeCatalyst espacio. Por ejemplo, CodeCatalyst los recursos incluyen un espacio o un proyecto. AWS los recursos incluyen un servicio Amazon ECS o una función Lambda. CloudTrail el registro debe configurarse por separado para cada Cuenta de AWS lugar en el que se implementen los recursos.

CodeCatalyst al iniciar sesión en las cuentas conectadas se incluyen las siguientes consideraciones:

- El acceso a CloudTrail los eventos se gestiona con IAM en la cuenta conectada y no en CodeCatalyst ella.
- Las conexiones de terceros, como la vinculación a un GitHub repositorio, harán que los nombres de los recursos de terceros se registren en CloudTrail los registros.

Note

CloudTrail El registro de CodeCatalyst eventos se realiza a nivel de espacio y no aísla los eventos según los límites del proyecto.

Para obtener más información al respecto CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

Note

En esta sección se describe el CloudTrail registro de todos los eventos registrados en un CodeCatalyst espacio y a los Cuentas de AWS que están conectados CodeCatalyst. Además, para revisar todos los eventos registrados en un CodeCatalyst espacio, también puede utilizar el `aws codecatalyst list-event-logs` comando AWS CLI y. Para obtener más información, consulte [Acceder a los eventos registrados en CodeCatalyst](#).

CodeCatalyst eventos espaciales

Las acciones CodeCatalyst para administrar los recursos a nivel de espacio y de proyecto se registran en la cuenta de facturación del espacio. Para CloudTrail iniciar sesión en un CodeCatalyst espacio, los eventos se registran teniendo en cuenta las siguientes consideraciones.

- CloudTrail los eventos se aplican a todo el espacio y no se centran en un solo proyecto.
- Al conectar y Cuenta de AWS a un CodeCatalyst espacio, los eventos que se puedan registrar para las conexiones de cuentas se registrarán en él. Cuenta de AWS Una vez habilitada esta conexión, no podrá desactivarla.
- Al conectar un Cuenta de AWS CodeCatalyst espacio y designarlo como la cuenta de facturación del espacio, los eventos se registrarán en élCuenta de AWS. Una vez habilitada esta conexión, no podrá desactivarla.

Los eventos de los recursos a nivel de espacio y de proyecto solo se registran en la cuenta de facturación. Para cambiar la cuenta de CloudTrail destino, actualice la cuenta de facturación en. CodeCatalyst Al comienzo del siguiente ciclo de facturación mensual, el cambio se aplicará a la nueva cuenta de facturación en CodeCatalyst. Después de eso, se actualiza la cuenta de CloudTrail destino.

Los siguientes son ejemplos de eventos relacionados con acciones CodeCatalyst para administrar los recursos a nivel de espacio y de proyecto. AWS Las siguientes API se publican mediante el SDK y la CLI. Los eventos se registrarán en la cuenta Cuenta de AWS especificada como cuenta de facturación del CodeCatalyst espacio.

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)
- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

CodeCatalyst conexiones de cuentas y eventos de facturación

Los siguientes son ejemplos de eventos relacionados con acciones relacionadas con las conexiones CodeCatalyst de cuentas o la facturación: AWS

- `AcceptConnection`
- `AssociateIAMRoletoConnection`
- `DeleteConnection`
- `DissassociateIAMRolefromConnection`

- `GetBillingAuthorization`
- `GetConnection`
- `GetPendingConnection`
- `ListConnections`
- `ListIAMRolesforConnection`
- `PutBillingAuthorization`
- `RejectConnection`

CodeCatalyst información en CloudTrail

CloudTrail está habilitada Cuenta de AWS al crear esa cuenta y al crearla. Cuando lo conectas Cuenta de AWS a un CodeCatalyst espacio, los eventos de ese espacio que se registran en ese espacio Cuenta de AWS se CloudTrail registran en esa cuenta de AWS. Los eventos que se pueden registrar CodeCatalyst se registran como CloudTrail eventos en los CloudTrail registros de la cuenta conectada y en el historial de eventos de la CloudTrail consola, junto con otros AWS eventos que se pueden registrar en esa cuenta.

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 la realizó un usuario con su ID de creador. AWS
- Si la solicitud se realizó con credenciales de usuario AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [elemento `userIdentity` de CloudTrail](#).

Acceder a CloudTrail los eventos

Para tener un registro continuo de los eventos que se produzcan en su entornoCuenta de AWS, incluidos los eventos relacionados con su CodeCatalyst actividadCuenta de AWS, cree un sendero.

Un registro de seguimiento permite a CloudTrail que pueda enviar archivos de registro a un bucket de S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de S3 especificado. Además, 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](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recepción de archivos de CloudTrail registro de varias regiones](#) y [recepción de archivos de CloudTrail registro de varias cuentas](#)

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 solicitud única 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.

Ejemplo de evento de conexiones de CodeCatalyst cuentas en AWS

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `ListConnections` acción. En el caso de una Cuenta de AWS que esté conectada al espacio, `ListConnections` se utiliza para ver todas las conexiones de cuentas a las que CodeCatalyst se ha conectado Cuenta de AWS. El evento se registrará según lo Cuenta de AWS especificado y el valor de `arn` será el nombre de recurso de Amazon (ARN) del rol utilizado para la acción.

`accountId`

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAI44QH8DHBEXAMPLE",
    "arn": "role-ARN",
    "accountId": "account-ID",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```

    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "role-ARN",
        "accountId": "account-ID",
        "userName": "user-name"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-06T15:04:31Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-09-06T15:08:43Z",
  "eventSource": "account-ID",
  "eventName": "ListConnections",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.168.0.1",
  "userAgent": "aws-cli/1.18.147 Python/2.7.18 Linux/5.4.207-126.363.amzn2int.x86_64
botocore/1.18.6",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "account-ID",
  "eventCategory": "Management"
}

```

Ejemplo de evento de recurso de CodeCatalyst proyecto en AWS

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `CreateDevEnvironment` acción. Una cuenta de AWS que esté conectada al espacio y que sea la cuenta de facturación designada para el espacio se usa para eventos a nivel de proyecto en el espacio, como la creación de un entorno de desarrollo.

DebajouserIdentity, en el `accountId` campo, se encuentra el ID de cuenta del IAM Identity Center (432677196278) que aloja el conjunto de identidades de todas las identidades de AWS

Builder ID. Este ID de cuenta contiene la siguiente información sobre el CodeCatalyst usuario del evento.

- El `type` campo indica el tipo de entidad de IAM de la solicitud. En el CodeCatalyst caso de los eventos relacionados con el espacio y los recursos del proyecto, este valor es `IdentityCenterUser`. El `accountId` campo especifica la cuenta propietaria de la entidad que se utilizó para obtener las credenciales.
- El `userId` campo contiene el identificador de AWS Builder ID del usuario.
- El `identityStoreArn` campo contiene el ARN del rol para la cuenta y el usuario del almacén de identidades.

El `recipientAccountId` campo contiene el ID de cuenta de la cuenta de facturación del espacio, con un valor de ejemplo de 111122223333.

Para obtener más información, consulte el [elemento `userIdentity` de CloudTrail](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IdentityCenterUser",
    "accountId": "432677196278",
    "onBehalfOf": {
      "userId": "user-ID",
      "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7"
    },
    "credentialId": "ABCDefGhiJKLmn11Lmn_1AbCDEFGHijk-AaBCdEFGHIjKLmnOPqrs11abEXAMPLE"
  },
  "eventTime": "2023-05-18T17:10:50Z",
  "eventSource": "codecatalyst.amazonaws.com",
  "eventName": "CreateDevEnvironment",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.168.0.1",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0",
  "requestParameters": {
    "spaceName": "MySpace",
    "projectName": "MyProject",
    "ides": [{
      "runtime": "public.ecr.aws/q6e8p2q0/cloud9-ide-runtime:2.5.1",
      "name": "Cloud9"
    }],
  },
}
```

```
"instanceType": "dev.standard1.small",
"inactivityTimeoutMinutes": 15,
"persistentStorage": {
  "sizeInGiB": 16
}
},
"responseElements": {
  "spaceName": "MySpace",
  "projectName": "MyProject",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 "
},
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventCategory": "Management"
}
```

Note

En algunos casos, es posible que no se conozca el agente de usuario. En este caso, CodeCatalyst proporcionará un valor de Unknown en el userAgent campo del CloudTrail evento.

Consultando las rutas de tus CodeCatalyst eventos

Puede crear y administrar consultas para sus CloudTrail registros mediante una tabla de consultas en Amazon Athena. Para obtener más información sobre cómo crear una consulta, consulte [Consultas de AWS CloudTrail registros](#) en la Guía del usuario de Amazon Athena.

Acceder a los eventos registrados en CodeCatalyst

Cuando los usuarios realizan acciones en Amazon CodeCatalyst, estas acciones se registran como eventos. Puedes usarlo AWS CLI para ver los registros de eventos en un espacio en un período de tiempo específico. Puedes ver estos eventos para revisar las acciones realizadas en el espacio, incluidas la fecha y la hora de las acciones, el nombre del usuario que realizó la acción y la dirección IP en la que el usuario realizó la solicitud.

Note

Los eventos de administración de un CodeCatalyst espacio se registran en CloudTrail la cuenta de facturación conectada. Para obtener más información sobre los eventos de CodeCatalyst administración que registra CloudTrail, consulte [CodeCatalyst información en CloudTrail](#).

Para ver el registro de eventos de un espacio, debe haberlo instalado y configurado AWS CLI con un perfil y debe tener la función de administrador del espacio. CodeCatalyst Para obtener más información, consulte [Configuración para usar el AWS CLI con CodeCatalyst](#) y [Función de administrador de espacios](#).

Note

Para ver el registro de eventos que se producen en nombre de CodeCatalyst un usuario conectado Cuentas de AWS o para ver el registro de eventos relacionados con el espacio o los recursos del proyecto en la cuenta de facturación conectada, puede utilizar AWS CloudTrail. Para obtener más información, consulte [Registro de llamadas a la CodeCatalyst API en Connected Cuentas de AWS mediante AWS CloudTrail](#).

1. Abre un terminal o una línea de comandos y ejecuta el `aws codecatalyst list-event-logs` comando especificando:
 - El nombre del espacio con la `--space-name` opción.
 - La fecha y la hora en las que desea empezar a revisar los eventos, en el formato de hora universal coordinada (UTC), tal como se especifica en el [RFC 3339](#), con la opción. `--start-time`
 - La fecha y la hora en las que desea dejar de revisar los eventos, en el formato de hora universal coordinada (UTC), tal como se especifica en el [RFC 3339](#), con la opción. `--end-time`
 - (Opcional) El número máximo de resultados que se pueden devolver en una sola respuesta, con la opción. `--max-results` Si el número de resultados es mayor que el número que especificó, la respuesta incluirá un `nextToken` elemento que podrá usar para devolver los siguientes resultados.

- (Opcional) Limite los resultados a un tipo de evento específico que desee que se devuelva, con la `--event-name` opción.

En este ejemplo, se devuelven los eventos registrados en el espacio denominado *ExampleCorp* desde el *30 de noviembre de 2022* hasta el 1 de *diciembre de 2022* y se devuelven un máximo de *2* eventos en la respuesta.

```
aws codecatalyst list-event-logs --space-name ExampleCorp --start-time 2022-11-30
--end-time 2022-12-01 --event-name list-event-logs --max-results 2
```

2. Si los eventos ocurrieron en este período de tiempo, el comando devuelve resultados similares a los siguientes:

```
{
  "nextToken": "EXAMPLE",
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "eventName": "listEventLogs",
      "eventType": "AwsApiCall",
      "eventCategory": "MANAGEMENT",
      "eventSource": "manage",
      "eventTime": "2022-12-01T22:47:24.605000+00:00",
      "operationType": "READONLY",
      "userIdentity": {
        "userType": "USER",
        "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111"
        "userName": "MaryMajor"
      },
      "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "requestPayload": {
        "contentType": "application/json",
        "data": "{\"spaceName\":\"ExampleCorp\",\"startTime\":
\"2022-12-01T00:00:00Z\",\"endTime\":\"2022-12-10T00:00:00Z\",\"maxResults\":
\"2\"}"
      },
      "sourceIpAddress": "127.0.0.1",
      "userAgent": "aws-cli/2.9.0 Python/3.9.11 Darwin/21.3.0 exe/x86_64
prompt/off command/codecatalyst.list-event-logs"
    },
    {
```

```

    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
    "eventName": "createProject",
    "eventType": "AwsApiCall",
    "eventCategory": "MANAGEMENT",
    "eventSource": "manage",
    "eventTime": "2022-12-01T09:15:32.068000+00:00",
    "operationType": "MUTATION",
    "userIdentity": {
      "userType": "USER",
      "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111",
      "userName": "MaryMajor"
    },
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "requestPayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject\", \"display_name\":\"MyFirstProject\"}"
    },
    "responsePayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject\", \"display_name\":\"MyFirstProject\", \"id\":\"a1b2c3d4-5678-90ab-cdef-EXAMPLE4444\"}"
    },
    "sourceIpAddress": "192.0.2.23",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0) Gecko/20100101 Firefox/102.0"
  }
]
}

```

3. Vuelva a ejecutar el `list-event-logs` comando con la `--next-token` opción y el valor del token devuelto para recuperar el siguiente conjunto de eventos registrados que coincidan con la solicitud.

Cuotas de identidad, permisos y acceso en CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de identidad, permisos y acceso en Amazon CodeCatalyst. Para obtener más información sobre las cuotas en Amazon CodeCatalyst, consulta [Cuotas para CodeCatalyst](#).

Recurso	Información
Alias en CodeCatalyst	<p>Cualquier combinación de caracteres permitida de entre 3 y 100 caracteres de longitud y que comience por una letra. Caracteres válidos: A-Z, a-z y 0-9. Los alias no pueden:</p> <ul style="list-style-type: none"> • contener menos de 3 caracteres • contener espacios o alguno de los siguientes caracteres: ? ^ * [\ ~ :
Número máximo de invitaciones enviadas por un usuario por día	500
Número máximo de invitaciones enviadas a una dirección de correo electrónico por día	25
Número máximo de fichas de acceso personal (PAT) por usuario	100
Contraseñas en CodeCatalyst	<p>Cualquier combinación de caracteres permitida de entre 8 y 64 caracteres de longitud. Caracteres válidos: A-Z, a-z y 0-9. La contraseña puede incluir los siguientes caracteres no alfanuméricos: (~ ! @ # \$ % ^ & * _ - + = ` \ { } [] : ; " ' < > , . ? /)</p>
Los nombres PAT están en CodeCatalyst	Cualquier combinación de caracteres permitida de entre 1 y 100 caracteres
Tiempo hasta que caduque la invitación de un miembro del proyecto	Caduca después de 24 horas
Tiempo hasta que caduque la invitación de un miembro del espacio	Caduca después de 24 horas

Recurso	Información
Tiempo hasta que caduque la verificación de una dirección de correo electrónico	Caduca 10 minutos después del envío

Solución de problemas

Esta sección puede ayudarte a solucionar algunos problemas comunes que puedes encontrar al acceder a tu CodeCatalyst perfil de Amazon.

¿Problemas al registrarte

Es posible que tengas algunos problemas al registrarte. Tenemos algunas soluciones.

Mi dirección de correo electrónico ya está en uso

Si el correo electrónico que has introducido ya está en uso y lo reconoces como propio, es posible que ya tengas un perfil con nosotros. Inicia sesión con esta identidad existente. Si no eres el propietario del correo electrónico existente, regístrate con otro que no utilices.

No puedo completar la verificación de correo

Si no has recibido tu correo electrónico de verificación

1. Revise su carpeta de correo no deseado, correo basura y elementos eliminados.

Note

Este correo electrónico de verificación proviene de la dirección `no-reply@signin.aws` `ono-reply@login.awsapps.com`. Le recomendamos que configure su sistema de correo para que acepte los mensajes con estas direcciones de correo electrónico como remitente y no los trate como correo basura o spam.

2. Espera 5 minutos y actualiza tu bandeja de entrada. Comprueba de nuevo la carpeta de correo no deseado, correo basura y elementos eliminados.
3. Si sigues sin ver el correo electrónico de verificación, selecciona Reenviar código. Si ya has salido de esa página, reinicia tu flujo de trabajo para registrarte en [Amazon CodeCatalyst](#).

Mi contraseña no cumple con los requisitos mínimos

Por motivos de seguridad, la contraseña debe incluir entre 8 y 20 caracteres, tanto mayúsculas como minúsculas, y números.

¿Problemas al iniciar sesión

He olvidado mi contraseña

Siga los pasos de [He olvidado mi contraseña](#).

Mi contraseña no funciona

Debe cumplir estos requisitos siempre que establezca o cambie su contraseña:

- Las contraseñas distinguen entre mayúsculas y minúsculas.
- Las contraseñas deben tener una longitud de entre 8 y 64 caracteres, con letras mayúsculas y minúsculas, números y al menos un carácter no alfanumérico.
- No puedes reutilizar las tres últimas contraseñas .

No puedo habilitar la MFA

Para habilitar la MFA, añada uno o más dispositivos MFA a su perfil siguiendo los pasos que se indican en [Autenticación multifactorial \(MFA\) en Amazon CodeCatalyst](#).

No puedo añadir un dispositivo MFA

Si descubre que no puede añadir otro dispositivo MFA, es posible que haya alcanzado el límite de dispositivos MFA que puede registrar. Es posible que tenga que quitar un dispositivo MFA existente antes de añadir uno nuevo.

No puedo quitar un dispositivo MFA

Si tiene intención de deshabilitar la MFA, quite el dispositivo MFA siguiendo los pasos que se indican en [Eliminar un dispositivo MFA](#). Sin embargo, si desea mantener la MFA habilitada, debe añadir otro dispositivo MFA antes de intentar eliminar un dispositivo MFA existente. Para obtener más información acerca de la adición de dispositivos MFA, consulte [¿Cómo registrar un dispositivo para su uso con la autenticación multifactorial](#).

¿Problemas para cerrar sesión

No encuentro dónde cerrar sesión

En la esquina superior derecha de la página, selecciona Cerrar sesión.

Cerrar sesión no significa que se cierre mi sesión por completo

El sistema está diseñado para cerrar sesión inmediatamente, pero cerrar la sesión por completo puede llevar hasta una hora.

Aparece el error «No existe un rol» debido a un flujo de trabajo fallido

Problema: Tras crear un proyecto desde la aplicación web o desde un blueprint sin servidor, el flujo de trabajo falla y se produce el siguiente error:

CLIENT_ERROR: El rol no existe

Posible solución: después de configurar un rol de IAM con los permisos para ejecutar el flujo de trabajo y haber agregado el rol de IAM al YAML del flujo de trabajo, el flujo de trabajo sigue fallando porque es posible que sea necesario agregar el rol de IAM a la conexión de su cuenta. Añada la función de IAM a la conexión de la cuenta de su espacio, tal y como se detalla en la sección. [Añadir funciones de IAM a las conexiones de cuentas](#)

Aparece un error de rol debido a un flujo de trabajo fallido

Problema: Tras crear un proyecto desde la aplicación web o desde un blueprint sin servidor, el flujo de trabajo falla y se produce el siguiente error:

CLIENT_ERROR: El rol no está configurado correctamente o no existe

Posible solución: es posible que el espacio en el que se creó el proyecto necesite configurar una Cuenta de AWS conexión o completar una solicitud de conexión de cuenta. Si tu espacio ya tiene una Cuenta de AWS conexión activa, crea y añade un rol de IAM con permisos para ejecutar acciones de flujo de trabajo. Añada el rol de IAM a la conexión de su cuenta, tal y como se detalla en. [Añadir funciones de IAM a las conexiones de cuentas](#)

Posible solución: si el proyecto se creó sin especificar una conexión, la conexión de la cuenta debe estar asociada al entorno de implementación. Si su espacio ya tiene una Cuenta de AWS conexión activa y se ha agregado un rol de IAM, debe agregar la conexión de cuenta con el rol de IAM a su

entorno de despliegue, tal y como se detalla en. [Añadir las funciones de IAM y de conexión de la cuenta a su entorno de implementación](#)

Necesito actualizar la función de IAM en el flujo de trabajo de un proyecto

Si la Cuenta de AWS conexión está completamente configurada y la función de IAM se crea y se añade a la conexión de la cuenta, puedes actualizar la función de IAM en el flujo de trabajo de tu proyecto.

1. Elija la opción CI/CD y elija su flujo de trabajo. Selecciona el botón YAML.
2. Elija Editar.
3. En el `ActionRoleArn`: campo, sustituya el ARN del rol de IAM por el ARN del rol de IAM actualizado. Elija Validar.
4. Elija Commit (Confirmar).

El flujo de trabajo se inicia automáticamente si se encuentra en la sucursal principal. De lo contrario, para volver a ejecutar el flujo de trabajo, seleccione Ejecutar.

¿Cómo relleno un formulario de asistencia?

Puedes ir a [Amazon CodeCatalyst](#) o rellenar un formulario de [comentarios de Support](#). En la sección Solicitud de información, en Cómo podemos ayudarte, indica que eres CodeCatalyst cliente de Amazon. Proporcione la mayor cantidad de detalles posible para que podamos estudiar su problema de mejor manera posible.

Extensiones en CodeCatalyst

Amazon CodeCatalyst incluye extensiones que te ayudan a añadir funcionalidad e integrarte con productos ajenos a CodeCatalyst. Con las extensiones del CodeCatalyst catálogo, los equipos pueden personalizar sus experiencias en ellas CodeCatalyst.

Temas

- [Conceptos de extensiones](#)
- [Instalación y desinstalación de extensiones en CodeCatalyst](#)
- [Uso de GitHub repositorios en CodeCatalyst](#)
- [Uso de problemas de Jira en CodeCatalyst](#)

Conceptos de extensiones

Estos son algunos conceptos y términos que debe conocer al trabajar con extensiones CodeCatalyst.

Extensiones

Una extensión es un complemento que puedes instalar en tu CodeCatalyst espacio para añadir nuevas funcionalidades a tus proyectos e integrarlas con servicios externos CodeCatalyst. Las extensiones se pueden buscar e instalar desde el CodeCatalyst catálogo.

CodeCatalyst catálogo

El CodeCatalyst catálogo es una lista centralizada de todas las extensiones disponibles en CodeCatalyst. Puedes consultar el CodeCatalyst catálogo para encontrar extensiones que puedan mejorar la experiencia de tu equipo en áreas CodeCatalyst como el origen, los flujos de trabajo, etc.

Instalación y desinstalación de extensiones en CodeCatalyst


Una extensión es un complemento que puedes instalar en tu CodeCatalyst espacio para añadir nuevas funciones a tus proyectos e integrarlas con servicios externos. CodeCatalyst Las extensiones se pueden buscar e instalar desde el CodeCatalyst catálogo.

Temas

- [Instalación de una extensión](#)

- [Desinstalar una extensión](#)


Instalación de una extensión

Puede instalar extensiones para su CodeCatalyst espacio que añadan funcionalidad a los proyectos de ese espacio. Puede ver el CodeCatalyst catálogo seleccionando el icono del catálogo 

Important

Para instalar una extensión, debe iniciar sesión con una cuenta que tenga la función de administrador del espacio.

Para instalar una extensión del CodeCatalyst catálogo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Navega hasta el CodeCatalyst catálogo seleccionando el icono de catálogo  en el menú superior. Puede buscar extensiones o filtrarlas en función de las categorías.
4. (Opcional) Elige el nombre de la extensión para ver más detalles sobre la extensión, como los permisos que tendrá la extensión.
5. Elija Instalar. Revise los permisos que requiere la extensión y, si quiere continuar, seleccione Instalar de nuevo.

Tras instalar una extensión, verás la página de detalles de la extensión instalada. Navegue por las pestañas para obtener más información sobre la extensión. La página de detalles también es donde podrá realizar una configuración adicional de la extensión si es necesario.


Desinstalar una extensión

Puede desinstalar las extensiones que se instalaron anteriormente en su CodeCatalyst espacio. Al desinstalar una extensión, es posible que se eliminen los recursos relacionados con esa extensión de tu CodeCatalyst espacio o tus proyectos.

⚠ Important

Para desinstalar una extensión, debe iniciar sesión con una cuenta que tenga la función de administrador del espacio.

Para desinstalar una extensión de tu CodeCatalyst espacio

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realiza una de las siguientes acciones para ver una lista de las extensiones instaladas en tu espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.
4. Seleccione Configurar en la extensión que desee desinstalar.
5. Selecciona Desinstalar en la página de detalles de la extensión.
6. Revise la información del cuadro de diálogo Desinstalar la extensión. Siga las instrucciones y, a continuación, seleccione Desinstalar para desinstalar la extensión.

Uso de GitHub repositorios en CodeCatalyst

GitHub es un servicio basado en la nube que ayuda a los desarrolladores a almacenar y gestionar su código. La extensión GitHub repositorios te permite usar GitHub repositorios enlazados en los proyectos de Amazon CodeCatalyst . También puedes vincular GitHub repositorios al crear un nuevo proyecto. CodeCatalyst Para obtener más información, consulte [Crear un proyecto con un GitHub repositorio vinculado](#).

📘 Note

No puedes usar GitHub repositorios vacíos o archivados con los proyectos. CodeCatalyst La extensión de GitHub repositorios no es compatible con GitHub Enterprise Server.

Una vez que instales y GitHub configures la extensión de repositorios, podrás:

- Consulta tus GitHub repositorios en la lista de repositorios de origen en CodeCatalyst
- Almacene y gestione los archivos de definición de flujos de trabajo en sus repositorios. GitHub
- Cree, lea, actualice y elimine archivos almacenados en GitHub repositorios enlazados desde CodeCatalyst entornos de desarrollo
- El CodeCatalyst flujo de trabajo de inicio se ejecuta automáticamente cuando el código se envía a sus repositorios GitHub
- Usa los archivos fuente de tu GitHub repositorio enlazado en CodeCatalyst los flujos de trabajo
- Lee y ejecuta GitHub acciones en los CodeCatalyst flujos de trabajo

En los siguientes temas se explica cómo instalar y configurar la extensión de GitHub repositorios y cómo utilizar GitHub los repositorios una vez que estén vinculados a los proyectos. CodeCatalyst

Temas

- [Inicio rápido: Uso de GitHub repositorios en CodeCatalyst](#)
- [Administrar GitHub cuentas en CodeCatalyst](#)
- [Administrar GitHub repositorios en CodeCatalyst](#)
- [Visualización de GitHub repositorios enlazados en CodeCatalyst](#)
- [Uso de GitHub repositorios en CodeCatalyst flujos de trabajo](#)
- [Uso de restricciones de acceso a direcciones IP en GitHub Enterprise Cloud](#)
- [El bloqueo de las fusiones de solicitudes de GitHub extracción cuando los flujos de trabajo fallan](#)

Inicio rápido: Uso de GitHub repositorios en CodeCatalyst

Realiza los siguientes pasos para instalar la extensión de GitHub repositorios, conectarte a tu GitHub cuenta y vincular tus GitHub repositorios a tus proyectos existentes CodeCatalyst .

También puedes instalar la extensión de GitHub repositorios, conectarte a tu GitHub cuenta y vincular GitHub repositorios al crear un nuevo proyecto. CodeCatalyst Para obtener más información, consulte [Crear un proyecto con un GitHub repositorio vinculado](#).

Contenido

- [Paso 1: Instale la GitHub extensión desde el catálogo CodeCatalyst](#)

- [Paso 2: Conecta tu GitHub cuenta a tu CodeCatalyst espacio](#)
- [Paso 3: vincula tu GitHub repositorio a tu proyecto CodeCatalyst](#)
- [Sigüientes pasos](#)


Paso 1: Instale la GitHub extensión desde el catálogo CodeCatalyst

El primer paso para utilizar GitHub los repositorios CodeCatalyst es instalar la extensión de GitHub repositorios desde el CodeCatalyst catálogo. Para instalar la extensión, lleve a cabo los siguientes pasos y seleccione la extensión de GitHub repositorios.

Important

Como parte de la instalación y configuración de la extensión de GitHub repositorios, debe instalar una extensión en su GitHub cuenta. Para ello, debe ser el administrador de la GitHub cuenta y el administrador de CodeCatalyst Space.

Para instalar una extensión del CodeCatalyst catálogo

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Navega hasta el CodeCatalyst CodeCatalyst catálogo seleccionando el icono de catálogo  en el menú superior. Puede buscar GitHub repositorios o filtrar extensiones en función de las categorías.
4. (Opcional) Para ver más detalles sobre la extensión, como los permisos que tendrá la extensión, elige el nombre de la extensión del GitHub repositorio.
5. Elija Instalar. Revise los permisos que requiere la extensión y, si quiere continuar, seleccione Instalar de nuevo.

Tras instalar la extensión de GitHub repositorios, accederás a la página de detalles de la extensión de GitHub repositorios, donde podrás ver y gestionar GitHub las cuentas conectadas y los repositorios enlazados. GitHub

Paso 2: Conecta tu GitHub cuenta a tu CodeCatalyst espacio

Tras instalar la extensión de GitHub repositorios, el siguiente paso es conectar tu GitHub cuenta a tu CodeCatalyst espacio.

Important

Para conectar tu GitHub cuenta a tu CodeCatalyst espacio, debes ser tanto el administrador de la GitHub cuenta como el administrador del CodeCatalyst espacio.

Para conectar tu GitHub cuenta a CodeCatalyst

1. En la pestaña **GitHub Cuentas conectadas**, selecciona **Conectar GitHub cuenta** para ir al sitio externo GitHub.
2. Inicia sesión en tu GitHub cuenta con tus GitHub credenciales y, a continuación, elige la cuenta en la que quieres instalar Amazon CodeCatalyst.

Tip

Si ya has conectado tu GitHub cuenta a otro espacio, no se te pedirá que vuelvas a autorizarla. En su lugar, aparecerá un cuadro de diálogo que le preguntará dónde desea instalar la extensión si es miembro o colaborador en más de una GitHub cuenta, o la página de configuración de la CodeCatalyst aplicación Amazon si solo pertenece a una GitHub cuenta. Configura la aplicación para el acceso al repositorio que quieres permitir y, a continuación, selecciona **Guardar**. Si el botón **Guardar** no está activo en GitHub, realice un cambio en la configuración e inténtelo de nuevo.

3. Elige si quieres permitir el acceso CodeCatalyst a todos los repositorios actuales y futuros, o elige los GitHub repositorios específicos en los que quieres usarlos. CodeCatalyst La opción predeterminada son todos los GitHub repositorios de la cuenta. GitHub
4. Revise los permisos otorgados a y CodeCatalyst, a continuación, seleccione **Instalar**.

Tras conectarte a tu GitHub cuenta CodeCatalyst, podrás ver la cuenta conectada en la pestaña de **GitHub cuentas** de la página de detalles de la extensión de GitHub repositorios.

Paso 3: vincula tu GitHub repositorio a tu proyecto CodeCatalyst

El tercer y último paso para usar tus GitHub repositorios CodeCatalyst es vincular el repositorio al CodeCatalyst proyecto en el que quieres usarlo.

Important

Si bien puedes vincular un GitHub repositorio como colaborador, solo puedes desvincular un GitHub repositorio como administrador del espacio o administrador del proyecto. Para obtener más información, consulte [Desvincular un GitHub repositorio de un proyecto CodeCatalyst](#).

Para vincular un GitHub repositorio a un CodeCatalyst proyecto desde la página de detalles de la extensión de GitHub repositorios

1. En la pestaña GitHub Repositorios enlazados, selecciona Vincular GitHub repositorio.
2. En el menú desplegable de la GitHub cuenta, selecciona la GitHub cuenta que contiene el repositorio que quieres vincular.
3. En el menú desplegable del GitHub repositorio, selecciona el repositorio que deseas vincular a un CodeCatalyst proyecto.

Tip

Si el nombre del repositorio está atenuado, no puedes vincular ese repositorio porque ya se ha vinculado a otro proyecto del espacio.

4. En el menú desplegable del CodeCatalyst proyecto, elige el CodeCatalyst proyecto al que quieres vincular el GitHub repositorio.
5. Elija Vincular.

También puedes vincular tu GitHub repositorio a un proyecto desde los repositorios de código fuente. Para obtener más información, consulte [Administrar GitHub repositorios en CodeCatalyst](#).

Siguientes pasos

Tras instalar la extensión de GitHub repositorios, conectar tu GitHub cuenta y vincular tus GitHub repositorios a tus CodeCatalyst proyectos, podrás utilizarla en CodeCatalyst flujos de trabajo y

entornos de desarrollo. Para obtener más información, consulte [Uso de GitHub repositorios en CodeCatalyst flujos de trabajo](#) y [Creación de un entorno de desarrollo](#).

Administrar GitHub cuentas en CodeCatalyst

Para usar GitHub los repositorios en CodeCatalyst, debes conectar tu GitHub cuenta a tu CodeCatalyst espacio. Si ya no quieres usar los repositorios de una cuenta CodeCatalyst, puedes desconectarla. Cuando se desconecta una cuenta, los eventos de los repositorios de la cuenta no iniciarán las ejecuciones del flujo de trabajo y no podrás usar esos repositorios con CodeCatalyst los entornos de desarrollo.

Important

Para conectar o desconectar tu GitHub cuenta y tu CodeCatalyst espacio, debes ser tanto el administrador de la GitHub cuenta como el administrador del CodeCatalyst espacio.


Contenido

- [Conectar una GitHub cuenta a un CodeCatalyst espacio](#)
- [Conectar una GitHub cuenta a un CodeCatalyst espacio durante la creación del proyecto](#)
- [Desconecta una GitHub cuenta de un CodeCatalyst espacio](#)

Conectar una GitHub cuenta a un CodeCatalyst espacio

Realiza los siguientes pasos para conectar una GitHub cuenta a tu CodeCatalyst espacio.

Para conectar tu GitHub cuenta a CodeCatalyst

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realice una de las siguientes acciones para ver la extensión de GitHub repositorios instalada en su espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.

4. En GitHub los repositorios, elija Configurar.
5. En la pestaña GitHub Cuentas conectadas, selecciona Conectar GitHub cuenta para ir al sitio externo GitHub.
6. Inicia sesión en tu GitHub cuenta con tus GitHub credenciales y, a continuación, elige la cuenta en la que quieres instalar Amazon CodeCatalyst.

 Tip

Si ya has conectado una GitHub cuenta al espacio, no se te pedirá que vuelvas a autorizar. En su lugar, aparecerá un cuadro de diálogo que le preguntará dónde desea instalar la extensión si es miembro o colaborador en más de un GitHub espacio, o la página de configuración de la CodeCatalyst aplicación Amazon si solo pertenece a un GitHub espacio. Configure la aplicación para el acceso al repositorio que desee permitir y, a continuación, seleccione Guardar. Si el botón Guardar no está activo, realice un cambio en la configuración e inténtelo de nuevo.

7. Elige si quieres permitir el acceso CodeCatalyst a todos los repositorios actuales y futuros, o elige los GitHub repositorios específicos en los que quieres usarlos. CodeCatalyst La opción predeterminada son todos los GitHub repositorios del espacio. GitHub
8. Revise los permisos otorgados a y CodeCatalyst, a continuación, seleccione Instalar.

Tras instalar la extensión de GitHub repositorios, accederás a la página de detalles de la extensión de GitHub repositorios, donde podrás ver y gestionar GitHub las cuentas conectadas y los repositorios enlazados GitHub .

Conectar una GitHub cuenta a un CodeCatalyst espacio durante la creación del proyecto

Puedes conectar una GitHub cuenta a tu CodeCatalyst espacio al crear un nuevo CodeCatalyst proyecto. Para obtener más información, consulte [Crear un proyecto con un GitHub repositorio vinculado](#).

Desconecta una GitHub cuenta de un CodeCatalyst espacio


Realiza los siguientes pasos para desconectar una GitHub cuenta de tu CodeCatalyst espacio. Una vez desconectada una cuenta, los eventos de los repositorios de la cuenta no iniciarán las

ejecuciones del flujo de trabajo y no podrás usar esos repositorios con entornos de CodeCatalyst desarrollo.

Note

Para desconectar una GitHub cuenta, primero debes desvincular todos los GitHub repositorios enlazados de esa cuenta. Para obtener más información, consulte [Desvincular un GitHub repositorio de un proyecto CodeCatalyst](#).

Para desconectar una cuenta GitHub

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realice una de las siguientes acciones para ver la extensión de GitHub repositorios instalada en su CodeCatalyst espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.
4. En GitHub los repositorios, elija Configurar.
5. En la pestaña de GitHub cuentas, selecciona la GitHub cuenta que deseas desconectar.
6. Selecciona Desconectar GitHub cuenta.
7. En el cuadro de diálogo Desconectar, revise los efectos de desconectar la cuenta.
8. Escribe desconectar en el campo de entrada de texto y, a continuación, selecciona Desconectar.

Administrar GitHub repositorios en CodeCatalyst

Para utilizar GitHub los repositorios CodeCatalyst, primero debes vincular los repositorios a tu proyecto. CodeCatalyst Antes de poder vincular un GitHub repositorio, debes conectar la GitHub cuenta a la que pertenece el repositorio con tu CodeCatalyst espacio. Para obtener más información, consulte [Conectar una GitHub cuenta a un CodeCatalyst espacio](#).

Si ya no quieres usar el GitHub repositorio en CodeCatalyst, puedes desvincularlo de un CodeCatalyst proyecto. Cuando un repositorio está desvinculado, los eventos de ese repositorio no

iniciarán las ejecuciones del flujo de trabajo y no podrás usar ese repositorio con CodeCatalyst Dev Environments.

Contenido

- [Vincula un GitHub repositorio a un proyecto CodeCatalyst](#)
- [Vincula un GitHub repositorio a un CodeCatalyst proyecto durante la creación del proyecto](#)
- [Desvincular un GitHub repositorio de un proyecto CodeCatalyst](#)

Vincula un GitHub repositorio a un proyecto CodeCatalyst

Puedes usar GitHub repositorios vinculados en los flujos de trabajo, donde los eventos del GitHub repositorio vinculado inician flujos de trabajo que podrían crear, probar o implementar código, según la configuración del flujo de trabajo. Los archivos de configuración del flujo de trabajo para los flujos de trabajo que utilizan GitHub repositorios vinculados se almacenan en el repositorio vinculado GitHub . GitHub Los repositorios vinculados también se pueden usar con los entornos de desarrollo para crear, actualizar y eliminar archivos en el repositorio vinculado GitHub . Puedes vincular un GitHub repositorio a un CodeCatalyst proyecto desde la página de detalles de la extensión de GitHub repositorios o desde la vista de repositorios de código fuente del propio proyecto.

Note

Un GitHub repositorio solo se puede vincular a un CodeCatalyst proyecto de un espacio. No puedes vincular un repositorio que tenga el mismo nombre que otro repositorio del proyecto.

Para vincular un GitHub repositorio a un CodeCatalyst proyecto desde la página de detalles de la extensión de GitHub repositorios

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realice una de las siguientes acciones para ver la extensión de GitHub repositorios instalada en su espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.

- b. Selecciona el icono del catálogo



en el menú superior.

4. En GitHub los repositorios, elija Configurar.
5. En la pestaña GitHub Repositorios enlazados, selecciona Vincular GitHub repositorio.
6. En el menú desplegable de GitHub cuentas, selecciona la GitHub cuenta que contiene el repositorio que quieres vincular.
7. En el menú desplegable del GitHub repositorio, elige el repositorio que quieres vincular a un CodeCatalyst proyecto.

 Tip

Si el nombre del repositorio está atenuado, no puedes vincular ese repositorio porque ya se ha vinculado a otro proyecto del espacio.

8. (Opcional) Si no ves un GitHub repositorio en la lista de repositorios, es posible que no esté configurado para el acceso al repositorio en la CodeCatalyst aplicación Amazon en GitHub. Puedes configurar GitHub los repositorios en los que se puede usar CodeCatalyst la cuenta conectada.
 - a. Ve a tu [GitHub](#) cuenta, selecciona Configuración y, a continuación, selecciona Aplicaciones.
 - b. En la pestaña GitHub Aplicaciones instaladas, selecciona Configurar para la CodeCatalyst aplicación Amazon.
 - c. Realice una de las siguientes acciones para configurar el acceso a los GitHub repositorios a los que desee vincular: CodeCatalyst
 - Para proporcionar acceso a todos los repositorios actuales y futuros, selecciona Todos los repositorios.
 - Para proporcionar acceso a repositorios específicos, selecciona Seleccionar solo repositorios, selecciona el menú desplegable Seleccionar repositorios y, a continuación, elige un repositorio al que quieras permitir el enlace. CodeCatalyst


 Note

- No puedes usar repositorios vacíos o archivados GitHub con los proyectos. CodeCatalyst
- No puedes vincular un GitHub repositorio que tenga el mismo nombre que un repositorio de un proyecto. CodeCatalyst
- La extensión de GitHub repositorios no es compatible con los repositorios de GitHub Enterprise Server.

9. En el menú desplegable del CodeCatalyst proyecto, selecciona el CodeCatalyst proyecto al que quieres vincular el GitHub repositorio.
10. Elija Vincular.

Para vincular un GitHub repositorio a un CodeCatalyst proyecto desde la página de repositorios de origen de un proyecto

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su CodeCatalyst proyecto.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.
4. Selecciona Añadir repositorio y, a continuación, selecciona Vincular repositorio.
5. En el menú desplegable del proveedor de repositorios, selecciona GitHub.
6. En el menú desplegable de GitHub cuentas, selecciona la GitHub cuenta que contiene el repositorio que quieres vincular.
7. En el menú desplegable del GitHub repositorio, elige el GitHub repositorio al que quieres vincular tu CodeCatalyst proyecto.


 Tip

Si el nombre del repositorio está atenuado, no puedes vincular ese repositorio porque ya se ha vinculado a otro proyecto en Amazon CodeCatalyst.

8. (Opcional) Si no ves un GitHub repositorio en el menú desplegable del GitHub repositorio, es posible que no esté configurado para el acceso al repositorio en el espacio en el que se

encuentra la aplicación. GitHub Puedes configurar GitHub los repositorios en los que se puede usar CodeCatalyst la cuenta conectada.

- a. Ve a tu [GitHub](#) cuenta, selecciona Configuración y, a continuación, selecciona Aplicaciones.
- b. En la pestaña GitHub Aplicaciones instaladas, selecciona Configurar para la CodeCatalyst aplicación Amazon.
- c. Realice una de las siguientes acciones para configurar el acceso a los GitHub repositorios a los que desee vincular: CodeCatalyst
 - Para proporcionar acceso a todos los repositorios actuales y futuros, selecciona Todos los repositorios.
 - Para proporcionar acceso a repositorios específicos, selecciona Seleccionar solo repositorios, selecciona el menú desplegable Seleccionar repositorios y, a continuación, elige un repositorio al que quieras permitir el enlace. CodeCatalyst

 Note

- No puedes usar repositorios vacíos o archivados GitHub con los proyectos. CodeCatalyst
- No puedes vincular un GitHub repositorio que tenga el mismo nombre que un repositorio de un proyecto. CodeCatalyst
- La extensión de GitHub repositorios no es compatible con los repositorios de GitHub Enterprise Server.

9. Elija Vincular.

Vincula un GitHub repositorio a un CodeCatalyst proyecto durante la creación del proyecto

Puede vincular un GitHub repositorio a un nuevo CodeCatalyst proyecto al crear el nuevo CodeCatalyst proyecto. Para obtener más información, consulte [Crear un proyecto con un GitHub repositorio vinculado](#).


Desvincular un GitHub repositorio de un proyecto CodeCatalyst

Al desvincular un repositorio no se elimina el GitHub repositorio ni se realiza ningún cambio en él. No elimina ningún archivo de configuración del flujo de trabajo almacenado en ese repositorio vinculado. Sin embargo, una vez que desvinculas un GitHub repositorio, los eventos de ese repositorio ya no iniciarán las ejecuciones del flujo de trabajo y no podrás usar el repositorio con entornos de desarrollo.

Important

Para desvincular tu GitHub repositorio de tu CodeCatalyst proyecto, debes ser el administrador del espacio o el administrador del proyecto.

Para desvincular un repositorio GitHub

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realice una de las siguientes acciones para ver la extensión de GitHub repositorios instalada en su espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.
4. En GitHub los repositorios, elija Configurar.
5. En la pestaña GitHub Repositorios enlazados, elige el GitHub repositorio que quieres desvincular.
6. Selecciona Desvincular el repositorio. GitHub
7. En el cuadro de diálogo Desvincular, revise los efectos de desvincular el repositorio.
8. Introduzca desvincular en el campo de entrada de texto y seleccione Desvincular.

Visualización de GitHub repositorios enlazados en CodeCatalyst

Puedes ver los repositorios enlazados en la lista de GitHub repositorios de origen de tu proyecto o en la página de detalles de la extensión de GitHub repositorios. Al seleccionarlos de la lista de

repositorios, no se abren. CodeCatalyst En su lugar, se abren en GitHub, donde puedes ver el código del repositorio vinculado y trabajar en él.

Para ver los GitHub repositorios enlazados en CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su CodeCatalyst proyecto.
3. En el panel de navegación, elija Código y, a continuación, elija Repositorios de origen.

Para ver los GitHub repositorios enlazados desde tu página de detalles de la extensión

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Ve a tu CodeCatalyst espacio y, a continuación, selecciona la pestaña Extensiones instaladas.
3. En GitHub los repositorios, selecciona Configurar.
4. Selecciona la pestaña GitHub Repositorios enlazados para ver todos los GitHub repositorios conectados a los CodeCatalyst proyectos de tu espacio. CodeCatalyst

Los GitHub repositorios que están vinculados a tu proyecto se muestran en la lista. Elija el GitHub repositorio en el que desee ver y editar los archivos. GitHub

Note

Si un flujo de trabajo utiliza un GitHub repositorio en una acción de origen, los cambios que realices en el flujo de trabajo YAML en el editor visual o en el editor YAML se CodeCatalyst confirmarán automáticamente y se enviarán al GitHub repositorio.

Uso de GitHub repositorios en CodeCatalyst flujos de trabajo

Puedes usar un GitHub repositorio vinculado como fuente de un flujo de trabajo, de modo que los cambios en una rama específica de un GitHub repositorio vinculado inicien automáticamente la ejecución de un flujo de trabajo.

Un flujo de trabajo es un procedimiento automatizado que describe cómo crear, probar e implementar el código como parte de un sistema de integración y entrega continuas (CI/CD). Un flujo de trabajo define una serie de pasos o acciones que se deben realizar durante la ejecución de un flujo de

trabajo. Un flujo de trabajo también define los eventos, o desencadenantes, que provocan el inicio del flujo de trabajo. Para configurar un flujo de trabajo, debe crear un archivo de definición del flujo de trabajo mediante el [editor visual o YAML](#) de la CodeCatalyst consola.

Tip

Para ver rápidamente cómo puedes usar los flujos de trabajo en un proyecto, [crea un proyecto con un plano](#). Cada esquema implementa un flujo de trabajo funcional que puede revisar, ejecutar y experimentar.

Al configurar un flujo de trabajo para usar un GitHub repositorio vinculado, el archivo de configuración del flujo de trabajo se almacena en ese GitHub repositorio. La configuración del flujo de trabajo es un archivo YAML que define el nombre, los activadores, los recursos, los artefactos y las acciones del flujo de trabajo. Para obtener más información sobre el archivo de configuración del flujo de trabajo, consulte [Referencia de definición de flujo de trabajo](#).

El archivo de configuración del flujo de trabajo debe estar en el `./codecatalyst/workflows/` directorio del GitHub repositorio.

Puede usar el editor de flujos de trabajo para crear y configurar flujos de trabajo. Para obtener más información, consulte [Introducción a los flujos de trabajo en CodeCatalyst](#) y [Trabajar con fuentes](#).

Inicie automáticamente la ejecución de un flujo de trabajo después de los eventos GitHub del repositorio

Puedes configurar un CodeCatalyst flujo de trabajo para que inicie automáticamente una ejecución cuando el código se envíe a la rama especificada de tu GitHub repositorio. Para iniciar la ejecución automática de un flujo de trabajo, añade un activador a la Triggers sección del archivo de configuración del flujo de trabajo.

Ejemplo: un simple disparador de pulsación de código

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se envía código a cualquier rama del repositorio de código fuente.

```
Triggers:  
- Type: PUSH
```

Ejemplo: un sencillo disparador de solicitudes de extracción

El siguiente ejemplo muestra un activador que inicia la ejecución de un flujo de trabajo cada vez que se crea una solicitud de extracción en cualquier rama del repositorio de origen.

```
Triggers:  
- Type: PULLREQUEST  
  Events:  
  - OPEN
```

Para obtener más información, consulte [Trabajar con factores desencadenantes](#).

Uso de restricciones de acceso a direcciones IP en GitHub Enterprise Cloud

La extensión Amazon CodeCatalyst GitHub repositories es compatible con las [restricciones de acceso IP de GitHub Enterprise Cloud](#). Al configurar una organización de GitHub Enterprise Cloud para restringir el acceso a direcciones IP específicas, también puedes [permitir que GitHub las aplicaciones configuren la lista](#) de direcciones IP permitidas, lo que permitirá CodeCatalyst registrar sus direcciones IP automáticamente. GitHub Como alternativa, puede [añadir las direcciones CodeCatalyst IP manualmente](#). Para obtener más información, consulte [Direcciones IP utilizadas por la extensión de GitHub repositorios](#).

Si las direcciones CodeCatalyst IP no están en la lista de direcciones permitidas de una GitHub organización, la CodeCatalyst GitHub aplicación Amazon no podrá acceder a tus GitHub repositorios.

Direcciones IP utilizadas por la extensión de GitHub repositorios

La extensión de GitHub repositorios utiliza las siguientes direcciones IP para acceder a los recursos de su GitHub organización:

```
us-west-2  
  52.32.242.246  
  54.148.176.49  
  35.164.118.94  
eu-west-1  
  34.241.64.10  
  34.246.255.80  
  3.248.38.7
```

El bloqueo de las fusiones de solicitudes de GitHub extracción cuando los flujos de trabajo fallan

Tras vincular un GitHub repositorio a CodeCatalyst, puedes añadir CodeCatalyst flujos de trabajo para las solicitudes de extracción. Se pueden ejecutar uno o más flujos de trabajo en una confirmación específica, y el estado de ejecución de cada flujo de trabajo también CodeCatalyst se refleja como parte del estado de confirmación en GitHub. Cuando se envía una nueva confirmación, los [estados de ejecución](#) del nuevo flujo de trabajo se reflejan en GitHub esa nueva confirmación. Si vuelves a ejecutar un flujo de trabajo para una confirmación, el estado de ejecución del nuevo flujo de trabajo anula el estado anterior de esa confirmación y flujo de trabajo.

Puedes configurar reglas de protección de ramificaciones GitHub para bloquear una fusión de solicitudes de incorporación de cambios cuando la última confirmación muestre un estado de ejecución de flujo de trabajo fallido. Con las reglas de protección de sucursales, el estado de la última confirmación afecta a la posibilidad de fusionar una solicitud de incorporación de cambios GitHub. Para obtener más información, consulta GitHub la documentación [Acerca de las comprobaciones de estado](#) y [Acerca de las sucursales protegidas](#). Para obtener más información sobre los flujos de trabajo, consulte [Trabajar con corridas](#) y [Trabajar con factores desencadenantes](#).

Uso de problemas de Jira en CodeCatalyst

Jira es una aplicación de software que ayuda a los equipos de desarrollo ágiles a planificar, asignar, realizar el seguimiento, elaborar informes y gestionar el trabajo. La extensión de Jira Software te permite usar proyectos de Jira en proyectos de Amazon CodeCatalyst .

Note

CodeCatalyst solo es compatible con Jira Software Cloud.

Una vez que instales y configures la extensión de Jira Software para un CodeCatalyst proyecto de Amazon, podrás:

1. Accede a los proyectos de Jira desde CodeCatalyst vinculándolos a otros proyectos CodeCatalyst
2. Actualiza los problemas de Jira con CodeCatalyst las solicitudes de incorporación de cambios
3. Consulta el estado y el flujo de trabajo de las solicitudes de CodeCatalyst extracción vinculadas en los problemas de Jira

En los siguientes temas se explica cómo instalar y configurar la extensión de Jira Software para usar Jira en ella. CodeCatalyst

Temas

- [Inicio rápido: problemas con el uso de Jira en CodeCatalyst](#)
- [Administrar sitios de Jira en CodeCatalyst](#)
- [Administrar proyectos de Jira en CodeCatalyst](#)
- [Vincular una incidencia de Jira a una solicitud de CodeCatalyst cambios](#)
- [Visualización de CodeCatalyst eventos en Jira](#)
- [Búsqueda de problemas de Jira en CodeCatalyst](#)

Inicio rápido: problemas con el uso de Jira en CodeCatalyst

Realiza los siguientes pasos para instalar la extensión de Jira Software, conectar tus sitios de Jira a tus CodeCatalyst espacios y vincular tus proyectos de Jira a tus proyectos. CodeCatalyst

Contenido

- [Paso 1: instala la extensión de Jira Software del catálogo CodeCatalyst](#)
- [Paso 2: Conecta tus sitios de Jira a tu espacio CodeCatalyst](#)
- [Paso 3: vincula tu proyecto de Jira a tu proyecto CodeCatalyst](#)
- [Paso 4: Vincular tus ediciones de Jira a una solicitud de incorporación de cambios CodeCatalyst](#)
- [Sigüientes pasos](#)


Paso 1: instala la extensión de Jira Software del catálogo CodeCatalyst

El primer paso para gestionar Jira CodeCatalyst es instalar la extensión de Jira Software del catálogo. CodeCatalyst Para instalar la extensión, realiza los siguientes pasos y elige la extensión de Jira Software.

Important

Como parte de la instalación y configuración de la extensión de Jira Software, debes tener una cuenta de Atlassian y un sitio de Jira existente, pero puedes crear un proyecto de Jira más adelante. Debes ser el administrador del sitio de Jira y el administrador de Space. CodeCatalyst

Para instalar una extensión del catálogo CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Navega hasta el CodeCatalyst CodeCatalyst catálogo seleccionando el icono del catálogo  en el menú superior. Puedes buscar Jira Software o filtrar las extensiones en función de las categorías.
4. (Opcional) Para ver más detalles sobre la extensión, como los permisos que tendrá, elige el nombre de la extensión de Jira Software.
5. Elija Instalar. Revise los permisos que requiere la extensión y, si quiere continuar, seleccione Instalar de nuevo.

Tras instalar la extensión de Jira Software, accederás a la página de detalles de la extensión de Jira Software, donde podrás ver y gestionar los sitios de Jira conectados y los proyectos de Jira enlazados.

Paso 2: Conecta tus sitios de Jira a tu espacio CodeCatalyst

Tras instalar la extensión de Jira Software, el siguiente paso es conectar tu sitio de Jira a tu espacio CodeCatalyst


Important

Para conectar tu sitio de Jira a tu CodeCatalyst espacio, debes ser tanto el administrador del sitio de Jira como el administrador del espacio CodeCatalyst

Para conectar tu sitio de Jira a CodeCatalyst

1. En la pestaña Connected Jira sites, selecciona Connect Jira site para ir al sitio externo de Atlassian Marketplace.
2. Selecciona Consíguelo ahora para empezar a instalar CodeCatalyst en tu sitio de Jira.
3. En función de su función, realice una de las siguientes acciones:

1. Si eres administrador de un sitio de Jira, en el menú desplegable del sitio, selecciona el sitio de Jira para instalar la CodeCatalyst aplicación y, a continuación, selecciona Instalar aplicación.

 Note


Si tienes un sitio de Jira, este paso no aparecerá y pasarás automáticamente al siguiente paso.

2. a. Si no eres administrador de Jira, en el menú desplegable del sitio, selecciona el sitio de Jira para instalar la CodeCatalyst aplicación y, a continuación, selecciona Solicitar aplicación. Para obtener más información sobre la instalación de aplicaciones de Jira, consulta [¿Quién puede instalar aplicaciones?](#) .
b. Introduce el motivo por el que necesitas realizar la instalación CodeCatalyst en el campo de texto o mantén el texto predeterminado y, a continuación, selecciona Enviar solicitud.
4. Revisa las acciones realizadas CodeCatalyst durante la instalación de la aplicación y, a continuación, selecciona Consíguela ahora.
5. Una vez instalada la aplicación, selecciona Volver a CodeCatalyst para volver a CodeCatalyst.

Tras conectar tu sitio de Jira CodeCatalyst, podrás verlo en la pestaña Sitios de Jira conectados de la página de detalles de la extensión de Jira Software.

Paso 3: vincula tu proyecto de Jira a tu proyecto CodeCatalyst

El tercer paso para gestionar tu proyecto de Jira CodeCatalyst consiste en vincular el proyecto de Jira al CodeCatalyst proyecto en el que quieres usarlo.

 Note

Un CodeCatalyst proyecto solo se puede vincular a un proyecto de Jira. Un proyecto de Jira se puede vincular a varios CodeCatalyst proyectos.

⚠ Important

Para vincular tu proyecto de Jira a tu CodeCatalyst proyecto, debes ser el administrador de CodeCatalyst Space o el administrador del CodeCatalyst proyecto.

Para vincular un proyecto de Jira a un CodeCatalyst proyecto desde la página de detalles de la extensión de Jira Software

1. En la pestaña Proyectos de Jira enlazados, selecciona Vincular un proyecto de Jira.
2. En el menú desplegable del sitio de Jira, selecciona el sitio de Jira que contiene el proyecto que quieres vincular.
3. En el menú desplegable del proyecto de Jira, selecciona el proyecto que quieres vincular a un proyecto. CodeCatalyst
4. En el menú desplegable del CodeCatalyst proyecto, selecciona el CodeCatalyst proyecto que quieres vincular a un proyecto de Jira.
5. Elija Vincular.

Una vez que un proyecto de Jira se vincula a un CodeCatalyst proyecto, el acceso a CodeCatalyst los problemas se deshabilita por completo y los problemas del panel de CodeCatalyst navegación se sustituirán por un elemento de problemas de Jira que enlaza con el proyecto de Jira.

Paso 4: Vincular tus ediciones de Jira a una solicitud de incorporación de cambios CodeCatalyst

Después de vincular un proyecto de Jira a CodeCatalyst otro, puedes vincular las CodeCatalyst incidencias creando una solicitud de extracción y haciendo que se muestren como una propiedad de la solicitud de extracción.

ℹ Note

No puedes crear una solicitud de extracción sin un repositorio de origen con dos ramas en tu CodeCatalyst proyecto. Para obtener más información sobre las solicitudes de extracción, consulta Cómo [trabajar con solicitudes de incorporación de cambios en CodeCatalyst](#).

Para vincular las incidencias de Jira a una solicitud de CodeCatalyst cambios

1. En el panel de navegación, selecciona Código y, a continuación, selecciona Solicitudes de extracción.
2. Selecciona Crear solicitud de extracción para introducir los detalles de la solicitud de extracción.
3. En el menú desplegable del repositorio de origen, selecciona el repositorio de origen al que quieres vincular una solicitud de extracción.
4. En el menú desplegable de la rama Fuente, selecciona la rama que contiene los cambios que deseas revisar.
5. En el menú desplegable de la rama de destino, selecciona la sucursal en la que deseas combinar los cambios revisados.
6. En el campo de entrada de texto del título de la solicitud de extracción, introduce el título de tus solicitudes de extracción.
7. Selecciona Vincular las incidencias de Jira en el campo opcional, selecciona el menú desplegable y busca las incidencias de Jira que quieras añadir desde el proyecto de Jira vinculado.
8. Selecciona las incidencias de Jira que quieras añadir a la solicitud de incorporación de nuevos usuarios.
9. Selecciona Crear para crear la solicitud de extracción.

El estado resumido de la solicitud de extracción y el estado de los eventos del CodeCatalyst flujo de trabajo asociados se reflejan en tu número de Jira.

Siguientes pasos

Tras instalar la extensión de Jira Software, conectar tu sitio de Jira, vincular tus proyectos de Jira a tu CodeCatalyst proyecto y vincular una solicitud de extracción, las actualizaciones CodeCatalyst se reflejan en tu proyecto de Jira. Para obtener más información sobre la visualización de CodeCatalyst eventos en Jira, consulte. [Visualización de CodeCatalyst eventos en Jira](#)

Administrar sitios de Jira en CodeCatalyst

Para gestionar los proyectos de Jira CodeCatalyst, debes conectar tus sitios de Jira a tu espacio. CodeCatalyst Si ya no quieres usar los proyectos de un sitio de Jira CodeCatalyst, puedes desconectar ese sitio de Jira. Cuando se desconecta un sitio de Jira, los problemas de Jira de los

proyectos del sitio no estarán disponibles en los CodeCatalyst proyectos y CodeCatalyst Issues volverá a ser el proveedor de los problemas.

Important

Para conectar o desconectar tu sitio de Jira y tu CodeCatalyst espacio, debes ser tanto el administrador del sitio de Jira como el administrador del espacio. CodeCatalyst


Contenido


- [Conectar un sitio de Jira a un espacio CodeCatalyst](#)
- [Desconectar un sitio de Jira de un espacio CodeCatalyst](#)

Conectar un sitio de Jira a un espacio CodeCatalyst

Realiza los siguientes pasos para conectar un sitio de Jira a tu CodeCatalyst espacio.

Para conectar tu sitio de Jira a CodeCatalyst


1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realiza una de las siguientes acciones para ver la extensión de Jira Software instalada en tu espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.
4. En Jira Software, selecciona Configurar.
5. En la pestaña Connected Jira sites, selecciona Connect Jira site para ir al sitio externo de Atlassian Marketplace.
6. Selecciona Consíguelo ahora para empezar a instalar CodeCatalyst en tu sitio de Jira.

 Note

Si ya lo instalaste CodeCatalyst en tu sitio de Jira, recibirás una notificación. Selecciona Comenzar para ir al paso final.

7. En función de su función, realice una de las siguientes acciones:

1. Si eres administrador de un sitio de Jira, en el menú desplegable del sitio, selecciona el sitio de Jira para instalar la CodeCatalyst aplicación y, a continuación, selecciona Instalar aplicación.

 Note

Si tienes un sitio de Jira, este paso no aparecerá y pasarás automáticamente al siguiente paso.

2. a. Si no eres administrador de Jira, en el menú desplegable del sitio, selecciona el sitio de Jira para instalar la CodeCatalyst aplicación y, a continuación, selecciona Solicitar aplicación. Para obtener más información sobre la instalación de aplicaciones de Jira, consulta [¿Quién puede instalar aplicaciones?](#) .
 - b. Introduce el motivo por el que necesitas realizar la instalación CodeCatalyst en el campo de texto o mantén el texto predeterminado y, a continuación, selecciona Enviar solicitud.
8. Revisa las acciones realizadas CodeCatalyst durante la instalación de la aplicación y, a continuación, selecciona Consíguela ahora.
9. Una vez instalada la aplicación, selecciona Volver a CodeCatalyst para volver a CodeCatalyst.

Tras conectar tu sitio de Jira CodeCatalyst, podrás verlo en la pestaña Sitios de Jira conectados de la página de detalles de la extensión de Jira Software.


Desconectar un sitio de Jira de un espacio CodeCatalyst

Realiza los siguientes pasos para desconectar un sitio de Jira de tu espacio. CodeCatalyst

Note

Para desconectar un sitio de Jira, primero debes desvincular todos los proyectos de Jira vinculados de esa cuenta. Para obtener más información, consulte [Desvincular un proyecto de Jira de otro proyecto CodeCatalyst](#).

Para desconectar un sitio de Jira

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.
3. Realiza una de las siguientes acciones para ver la extensión de Jira Software instalada en tu espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo  en el menú superior.
4. En Jira Software, selecciona Configurar.
5. En la pestaña Sitios de Jira conectados, elige el sitio de Jira que quieres desconectar.
6. Selecciona Desconectar el sitio de Jira.
7. En el cuadro de diálogo Desconectar, revisa los efectos de desconectar el sitio.
8. Escribe desconectar en el campo de entrada de texto y, a continuación, selecciona Desconectar.

Administrar proyectos de Jira en CodeCatalyst

Para gestionar un proyecto de Jira CodeCatalyst, primero debes vincular el proyecto de Jira al tuyo. CodeCatalyst Luego, puedes añadir y usar los problemas de Jira a una solicitud de incorporación de cambios en un CodeCatalyst proyecto. CodeCatalyst Para poder vincular un proyecto de Jira, debes conectar el sitio de Jira al que pertenece el proyecto con tu espacio. CodeCatalyst Para obtener más información, consulte [Conectar un sitio de Jira a un espacio CodeCatalyst](#).

Si ya no quieres usar un proyecto de Jira CodeCatalyst, puedes desvincularlo de tu proyecto. CodeCatalyst Cuando se desvincula un proyecto de Jira, los problemas de Jira no estarán

disponibles en el CodeCatalyst proyecto y CodeCatalyst Issues volverá a ser el proveedor de los problemas.

Important

Para vincular tu proyecto de Jira a tu CodeCatalyst proyecto, debes ser el administrador de CodeCatalyst Space o el administrador del proyecto. CodeCatalyst

Contenido

- [Vincular un proyecto de Jira a un proyecto CodeCatalyst](#)
- [Desvincular un proyecto de Jira de otro proyecto CodeCatalyst](#)

Vincular un proyecto de Jira a un proyecto CodeCatalyst

Puedes usar proyectos de Jira vinculados para gestionar las incidencias y vincular las solicitudes de incorporación de cambios a una incidencia de CodeCatalyst Jira. El estado resumido de una solicitud de cambios y el estado de los eventos del CodeCatalyst flujo de trabajo asociados se reflejan en tu edición de Jira.

Note

Un CodeCatalyst proyecto solo se puede vincular a un proyecto de Jira. Un proyecto de Jira se puede vincular a varios CodeCatalyst proyectos.

Para vincular un proyecto de Jira a un CodeCatalyst proyecto desde la página de detalles de la extensión de Jira Software

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navega hasta tu CodeCatalyst espacio.
3. Realiza una de las siguientes acciones para ver la extensión de Jira Software instalada en tu espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.

- b. Selecciona el icono del catálogo



en el menú superior.

4. En Jira Software, selecciona Configurar.
5. En la pestaña Proyectos de Jira vinculados, selecciona Vincular el proyecto de Jira.
6. En el menú desplegable del sitio de Jira, selecciona el sitio de Jira que contiene el proyecto que quieres vincular.
7. En el menú desplegable del proyecto de Jira, selecciona el proyecto que quieres vincular a un proyecto. CodeCatalyst
8. En el menú desplegable del CodeCatalyst proyecto, selecciona el CodeCatalyst proyecto que quieres vincular a un proyecto de Jira.
9. Elija Vincular.

Una vez que un proyecto de Jira se vincula a un CodeCatalyst proyecto, el acceso a CodeCatalyst los problemas se desactiva por completo y los problemas del panel de CodeCatalyst navegación se sustituirán por un elemento de problemas de Jira que enlaza con el proyecto de Jira.

Desvincular un proyecto de Jira de otro proyecto CodeCatalyst

Al desvincular un proyecto, no se elimina el proyecto de Jira, incluidos los elementos de planificación o la información de desarrollo, ni se realiza ningún cambio en él. Sin embargo, una vez que desvincules un proyecto de Jira, los problemas de Jira del proyecto dejarán de estar disponibles para vincularlos al proyecto. CodeCatalyst

Para desvincular un proyecto de Jira

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navega hasta tu CodeCatalyst espacio.
3. Realiza una de las siguientes acciones para ver la extensión de Jira Software instalada en tu espacio:
 - a. Selecciona Configuración y, a continuación, selecciona Extensiones instaladas.
 - b. Selecciona el icono del catálogo



en el menú superior.

4. En la pestaña Proyectos de Jira vinculados, selecciona el proyecto de Jira que quieres desvincular.
5. Selecciona Desvincular el proyecto de Jira.
6. En el cuadro de diálogo Desvincular, revise los efectos de desvincular el repositorio.
7. Introduzca desvincular en el campo de entrada de texto y seleccione Desvincular.

Vincular una incidencia de Jira a una solicitud de CodeCatalyst cambios

Puedes vincular las solicitudes de cambios que se crean en un repositorio CodeCatalyst fuente a las ediciones de Jira. Tras vincular una incidencia de Jira, la incidencia se muestra como una propiedad de la solicitud de cambios. Como resultado, los eventos de solicitud de extracción, de flujo de trabajo y de despliegue se envían a Jira y se añaden a la edición de Jira. Las solicitudes de extracción se pueden vincular a uno o varios problemas de Jira. Solo puedes vincular las solicitudes de cambios que estén en un repositorio de CodeCatalyst origen, no las que estén en un repositorio de terceros. GitHub Para poder vincular las incidencias de Jira a una solicitud de extracción, tu proyecto de Jira debe estar vinculado al CodeCatalyst proyecto. Para obtener más información sobre cómo vincular un proyecto de Jira a un CodeCatalyst proyecto, consulta [Administrar sitios de Jira en CodeCatalyst](#)

Note

No puedes crear una solicitud de incorporación de cambios sin un repositorio de código fuente con dos ramas en tu CodeCatalyst proyecto. Para obtener más información sobre las solicitudes de extracción, consulta [Cómo trabajar con solicitudes de extracción en CodeCatalyst](#).

Para vincular las incidencias de Jira a una solicitud de CodeCatalyst cambios

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su CodeCatalyst proyecto.
3. En el panel de navegación, selecciona Código y, a continuación, selecciona Solicitudes de extracción.
4. Selecciona Crear solicitud de extracción para introducir los detalles de la solicitud de extracción.
5. En el menú desplegable del repositorio de origen, selecciona el repositorio de origen al que quieres vincular una solicitud de extracción.

6. En el menú desplegable de la rama Fuente, selecciona la rama que contiene los cambios que deseas revisar.
7. En el menú desplegable de la rama de destino, selecciona la sucursal en la que deseas combinar los cambios revisados.
8. En el campo de entrada de texto del título de la solicitud de extracción, introduce el título de tus solicitudes de extracción.
9. Selecciona Vincular las incidencias de Jira en el campo opcional, selecciona el menú desplegable y busca las incidencias de Jira que quieras añadir desde el proyecto de Jira vinculado.
10. Selecciona las incidencias de Jira que quieras añadir a la solicitud de extracción.
11. Selecciona Crear para crear la solicitud de extracción.

Una vez que vincules las incidencias de Jira a una solicitud de CodeCatalyst extracción, aparecerá un resumen de la solicitud de extracción. El resumen incluye la ejecución del flujo de trabajo, las cuestiones vinculadas, los revisores obligatorios, los revisores opcionales y el autor.

Note

La información asociada al problema de Jira no está disponible en CodeCatalyst

Tras vincular una solicitud de incorporación de cambios, el CodeCatalyst proyecto sincronizado y el proyecto de Jira permiten que las actualizaciones se reflejen en tu proyecto de CodeCatalyst Jira. El estado de la solicitud de extracción vinculada y cualquier evento del flujo de trabajo relacionado con la solicitud de extracción aparecerán en la edición de Jira al visualizarla en Jira. Para obtener más información sobre la visualización de CodeCatalyst eventos en Jira, consulta [Visualización de CodeCatalyst eventos en Jira](#)

Visualización de CodeCatalyst eventos en Jira

Si tus CodeCatalyst proyectos y los proyectos de Jira están vinculados, el estado resumido de la solicitud de cambios y el estado de los eventos del CodeCatalyst flujo de trabajo asociados se reflejan en tu número de Jira. Por ejemplo, si cierras o fusionas una solicitud de extracción CodeCatalyst, la actualización de estado se refleja en la edición de Jira. CodeCatalyst Los eventos de CI/CD del flujo de trabajo relacionados con una solicitud de CodeCatalyst extracción están

sincronizados, por lo que si el flujo de trabajo se ejecuta correctamente, también se enviará a la edición de Jira.

Para ver los CodeCatalyst eventos de una edición de Jira

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navegue hasta su CodeCatalyst proyecto.
3. En el panel de CodeCatalyst navegación, selecciona Código, selecciona Solicitudes de extracción y, a continuación, elige la solicitud de extracción con el problema de Jira que quieres ver en tu proyecto de Jira.
4. En el panel Información adicional, selecciona el tema de Jira que quieres ver en tu proyecto de Jira.
5. En el panel de detalles del proyecto de Jira, selecciona la solicitud de extracción incluida en la lista de desarrollo para ver los detalles de la solicitud de extracción.
6. (Opcional) Para ver las versiones más recientes, selecciona la pestaña Compilaciones.
7. (Opcional) Para ver el estado de desarrollo, selecciona la pestaña Implementaciones.

Búsqueda de problemas de Jira en CodeCatalyst

Tras vincular un proyecto de Jira, puedes buscar problemas en el proyecto de Jira vinculado mediante la barra de búsqueda CodeCatalyst global. También puedes buscar las incidencias de Jira CodeCatalyst al vincular las incidencias de una solicitud de incorporación de cambios. Para obtener más información sobre cómo vincular las incidencias de Jira a una solicitud de incorporación de CodeCatalyst cambios, consulta. [Vincular una incidencia de Jira a una solicitud de CodeCatalyst cambios](#)

Para buscar problemas de Jira en proyectos de Jira vinculados

1. [Abre la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navegue hasta su CodeCatalyst proyecto.
3. En la barra de búsqueda global, busca en un proyecto de Jira vinculado las incidencias o las que quieras vincular a una solicitud de incorporación de cambios.

Buscar en CodeCatalyst

Usa la barra de búsqueda o una ventana de resultados de búsqueda específica CodeCatalyst para buscar código, problemas, proyectos y usuarios CodeCatalyst.

Para encontrar recursos en tu espacio y en tus proyectos, introduce consultas como el nombre, la descripción y el estado en la barra de búsqueda. También puedes restringir las consultas de búsqueda mediante el lenguaje de consulta de búsqueda.

Temas

- [Refinar tu consulta de búsqueda](#)
- [Consideraciones a la hora de trabajar con la búsqueda](#)
- [Referencia de campos en los que se pueden buscar](#)

Para buscar

1. En la barra de búsqueda de la barra de navegación superior, introduce una consulta de búsqueda.
2. (Opcional) Refina la consulta de búsqueda mediante CodeCatalyst el idioma de la consulta de búsqueda. Para obtener más información, consulte [Refinar tu consulta de búsqueda](#).
3. Haga una de las siguientes acciones:
 - Para buscar recursos dentro del proyecto en el que se encuentra actualmente, elija Este proyecto.
 - Para buscar recursos en todos los proyectos del espacio en el que se encuentra actualmente, elija Este espacio.
4. Para ver los resultados de la búsqueda en una ventana dedicada a los resultados de la búsqueda, realice una de las siguientes acciones:
 - En la parte inferior de la ventana de resultados de búsqueda rápida, selecciona Ver todos los resultados en nombre-proyecto | nombre-espacio para ver todos los resultados de la búsqueda.
 - Pulse Entrar para ver todos los resultados de la búsqueda.

i Tip

Menciona a otros usuarios del proyecto en un comentario o descripción de una solicitud de selección, o en un comentario o descripción de un tema, utilizando el signo @ seguido de su nombre visible o nombre de usuario. También puedes vincular a recursos como publicaciones o archivos de código utilizando el signo @ seguido del nombre de la publicación o del archivo de código.

Refinar tu consulta de búsqueda

Si no encuentras lo que buscas después de buscar, puedes afinar la búsqueda con un lenguaje CodeCatalyst de consulta especializado. Los campos individuales no tienen límite de caracteres, pero la consulta general tiene un límite de 1024 caracteres.

Temas

- [Refinar por tipo](#)
- [Refinar por campo](#)
- [Refinar con operadores booleanos](#)
- [Refinando por proyecto](#)

Refinar por tipo

Para limitar el alcance de la búsqueda a un tipo de información específico, incluye *type:result-type* en la búsqueda donde está el *tipo de resultado* `code`, `issue`, `project` o `user`.

Ejemplos:

- `type:code AND java`— Mostrar los resultados del código en los campos relacionados con el código que contengan «java».

Para obtener más información, consulte [Campos de código](#).

- `type:issue AND Bug`— Mostrar los resultados de los problemas en los campos relacionados con el problema que contengan la palabra «Error».

Para obtener más información, consulte [Campos de emisión](#).

- `type:user AND MaryMajor`— Mostrar los resultados de los usuarios en los campos relacionados con el usuario que contengan «». `MaryMajor`

Para obtener más información, consulte [Campos de usuario](#).

- `type:project AND Datafeeder`— Mostrar los resultados del proyecto que contengan «Datafeeder».

Para obtener más información, consulte [Campos del proyecto](#).

Refinar por campo

Para limitar el alcance de la búsqueda a un campo específico, incluya *field-name:query* en la búsqueda donde el *nombre del campo* es `titleusername,project,description,, etc.` y *consulta* es el texto que está buscando. Para obtener una lista de campos, consulte. [Referencia de campos en los que se pueden buscar](#) Puede buscar varias consultas mediante paréntesis.

Ejemplos:

- `title:bug`— Mostrar los resultados donde el título contiene la palabra «error».
- `username:John`— Mostrar los resultados en los que el nombre de usuario contenga «John».
- `project:DataFeeder`— Mostrar los resultados en el proyecto «DataFeeder». La consulta no distingue entre mayúsculas y minúsculas.
- `description:overview`— Mostrar los resultados donde la descripción contiene un «resumen».

Refinar con operadores booleanos

Para especificar las restricciones en las frases de búsqueda, puede utilizar los operadores booleanos, y. AND OR NOT Si incluyes varias frases, CodeCatalyst úsalas con ellas de forma OR predeterminada. Puedes agrupar las frases de búsqueda mediante paréntesis.

- `exception AND type:code`— Mostrar solo los resultados del código de «excepción».
- `path:README.md AND repo:ServerlessAPI`— Muestra los resultados de las rutas con «README.md» donde el repositorio se denomina «ServerlessAPI».
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)`— Muestra los resultados de «buildspec.yml», donde el repositorio es «ServerlessAPI» o «». `ServerlessWebApp`

- `path:java NOT (path:py OR path:ts)`—Mostrar los resultados en los que la ruta contiene «java» pero no «py» o «ts».

Refinando por proyecto

Para limitar el alcance de la búsqueda a un proyecto específico, *project:name AND query* inclúyala en la búsqueda, donde *nombre* es el proyecto en el que está buscando y *consulta* es el contenido que está buscando.

- `project:name AND query`—Mostrar los resultados donde la ruta contiene la consulta y el nombre del proyecto.

Consideraciones a la hora de trabajar con la búsqueda

Actualizaciones de contenido retrasadas: las actualizaciones de contenido, como los cambios de nombre o las reasignaciones de problemas, pueden tardar varios minutos en reflejarse en los resultados de la búsqueda. Las actualizaciones importantes, como la migración de una base de código, pueden tardar más en aparecer en los resultados de búsqueda.

Caracteres especiales de escape: los siguientes caracteres especiales requieren una consideración especial en las consultas de búsqueda: `+ - & & || ! () { } [] ^ " ~ * ? : \`. Los caracteres especiales no influirán en la consulta, y debes eliminarlos o evitarlos. Para escapar de un carácter, añade una barra invertida (`\`) delante de él. Por ejemplo, la consulta de búsqueda `[Característica]` debe ser `Característica o \ [Característica]`.

Limitar la búsqueda: la búsqueda no distingue entre mayúsculas y minúsculas. La búsqueda en minúsculas evita que las consultas dividan las palabras al cambiar entre mayúsculas y minúsculas. Por ejemplo, para realizar consultas por `MyService` y `soloMyService`, considere la posibilidad de realizar consultas `myservice` para evitar resultados que contengan únicamente `o. my service`

De forma predeterminada, la búsqueda une palabras y partes de palabras con una conjunción en sentido OR. Por ejemplo, `new function` podría devolver resultados que contengan ambas `new function` y también resultados que contengan únicamente `new ofunction`. Para evitar esto último, combine varias palabras con AND. Por ejemplo, puede buscar `new AND function`.

Ramas predeterminadas: la búsqueda solo devolverá los resultados de código de la última confirmación en la rama predeterminada de un repositorio de código fuente. Para encontrar código en otras ramas o confirmaciones, puedes [clonar el repositorio de forma local](#), [abrir la rama en un](#)

[entorno de desarrollo](#) o [ver las ramificaciones y los detalles en la CodeCatalyst interfaz](#) de usuario. Al cambiar la rama predeterminada, se actualizan los archivos que se pueden detectar mediante la búsqueda. Para obtener más información, consulte [Ver y cambiar la rama predeterminada de un repositorio](#).

Referencia de campos en los que se pueden buscar

CodeCatalyst busca en los siguientes campos al introducir consultas de búsqueda. Los alias son otro nombre que puede utilizar para hacer referencia al campo en el lenguaje de consultas avanzadas.

Campos de código

Campo	Alias	Descripción
Nombre de sucursal	branch	Nombre de la sucursal en la que se encuentra el archivo de códigos.
code	N/A	Información sobre el contenido del código en forma de fragmentos de código que indican las partes del código fuente que coinciden con la búsqueda.
ID de confirmación	N/A	ID de confirmación de la confirmación en la que se actualizó por última vez el archivo de código devuelto. Puede ser o no el ID de confirmación que aparece en la punta del nombre de la rama especificado en. <code>branchName</code>
Mensaje de confirmación	N/A	Mensaje de confirmación de la confirmación en la que se actualizó por última

Campo	Alias	Descripción
		vez el archivo de código. Puede ser o no el mensaje de confirmación que aparece en la punta del nombre de la rama especificado en. <code>branchName</code> Si no se proporcionó ningún mensaje de confirmación, este valor será una cadena vacía.
<code>filePath</code>	ruta	Ruta de este archivo de código.
<code>lastUpdatedBy</code>	N/A	CodeCatalyst usuario que actualizó el archivo de código por última vez. Si el nombre de usuario no está disponible, este valor será la dirección de correo electrónico del usuario tal como está configurada en el archivo de configuración de Git.
<code>lastUpdatedByID</code>	N/A	ID único generado por el sistema del usuario que actualizó por última vez el archivo de códigos. Si el ID de usuario no está disponible, este valor puede ser la dirección de correo electrónico del usuario.

Campo	Alias	Descripción
lastUpdatedTime	N/A	Hora en la que los datos de búsqueda se actualizaron por última vez con la confirmación que contenía el archivo de código (en la marca de tiempo universal coordinada (UTC)).
projectId	N/A	ID único del proyecto generado por el sistema.
projectName	Nombres de proyectos, proyecto	Muestra el nombre del proyecto que contiene el repositorio fuente en el que se ha archivado el archivo de código.
ID del repositorio	RepoID	ID único del repositorio de origen generado por el sistema.
Nombre del repositorio	repositorio, repositorio	Muestra el nombre del repositorio fuente en el que se ha archivado el archivo de código.

Campos de emisión

Campo	Alias	Descripción
ID de cesionario	ID de cesionario	Identificadores únicos generados por el sistema de los usuarios asignados al problema.

Campo	Alias	Descripción
cesionarios	cesionario	Nombres de usuario de los usuarios asignados a la cuestión.
Creado por	N/A	Muestra el nombre del usuario que creó la incidencia.
createdById	N/A	ID único generado por el sistema del usuario que creó el problema.
Hora de creación	N/A	Hora en que se creó el problema (en la marca de tiempo universal coordinada (UTC)).
description	N/A	Descripción del problema.
está archivado	archived	Valor booleano que indica si se debe crear el problema en un estado archivado.
Está bloqueado	blocked	Valor booleano que indica si el problema está marcado como bloqueado.
ID de etiqueta	ID de etiqueta	Identificadores únicos de las etiquetas de un número generados por el sistema.
lastUpdatedBy	N/A	Muestra el nombre del usuario que actualizó la edición por última vez.

Campo	Alias	Descripción
lastUpdatedByID	N/A	ID único generado por el sistema del usuario que actualizó el problema por última vez.
lastUpdatedTime	N/A	Hora en que se actualizó el problema por última vez (en la marca de tiempo universal coordinada (UTC)).
priority	N/A	Prioridad del problema, si se ha asignado alguna.
projectId	N/A	ID único del proyecto generado por el sistema.
projectName	Nombres de proyectos, proyecto	Proyecto en el que se encuentra este problema.
ShortID	N/A	Identificador abreviado y que se incrementa automáticamente para el problema.
status	N/A	Estado del problema que indica si el problema está pendiente o en una columna incluida.
ID de estado	N/A	Identificador del estado del sistema.
title	N/A	Título de la emisión.

Campos del proyecto

Campo	Alias	Descripción
description	N/A	Descripción del proyecto.
lastUpdatedTime	N/A	Hora en la que se actualizaron por última vez los metadatos del proyecto (en la marca de tiempo universal coordinada (UTC)).
projectName	project	Nombre del proyecto en el espacio.
Ruta del proyecto	N/A	Nombre del proyecto que se puede enrutar mediante URL, definido durante la creación del proyecto. Se utiliza en las direcciones URL que requieren el nombre del proyecto.

Campos de usuario

Campo	Alias	Descripción
displayName	N/A	Nombre utilizado para el usuario en CodeCatalyst. Los nombres para mostrar no son únicos.
email	N/A	Dirección de correo electrónico del usuario.
lastUpdatedTime	N/A	Hora en la que se actualizaron por última vez los metadatos

Campo	Alias	Descripción
		del usuario (en la marca de tiempo universal coordinada (UTC)).
userName	username	Nombre de usuario elegido por el usuario al registrarse. CodeCatalyst A diferencia de los nombres para mostrar, los nombres de usuario no se pueden cambiar.

Solución de problemas de Amazon CodeCatalyst

La siguiente información puede ayudarle a solucionar problemas comunes en CodeCatalyst. También puedes usar el informe de CodeCatalyst salud de Amazon para determinar si hay problemas de servicio que puedan afectar a tu experiencia.

Temas

- [Solución de problemas generales de acceso](#)
- [Solución de problemas de soporte](#)
- [Parte o la totalidad de Amazon CodeCatalyst no está disponible](#)
- [No puedo crear un proyecto en CodeCatalyst](#)
- [Quiero enviar comentarios en CodeCatalyst](#)
- [Solución de problemas con los repositorios de código fuente](#)
- [Solución de problemas de proyectos y planos](#)
- [Solución de problemas con los flujos de trabajo](#)
- [Solución de problemas con la búsqueda en CodeCatalyst](#)
- [Solución de problemas con las cuentas asociadas a tu espacio](#)
- [Solución de problemas con entornos de desarrollo](#)
- [Solución de problemas con problemas](#)
- [Solución de problemas entre Amazon CodeCatalyst y los AWS SDK o el AWS CLI](#)

Solución de problemas generales de acceso

He olvidado mi contraseña

Problema: he olvidado la contraseña que utilizo para mi ID de AWS constructor y Amazon CodeCatalyst.

Posibles soluciones: la forma más sencilla de solucionar este problema es restablecer la contraseña.

1. Abre [Amazon CodeCatalyst](#) e introduce tu dirección de correo electrónico. Después, elija Continue (Continuar).
2. Elige ¿Has olvidado tu contraseña?

3. Te enviaremos un correo electrónico con un enlace para que cambies tu contraseña. Si no ves el correo electrónico en tu bandeja de entrada, revisa tu carpeta de correo no deseado.

Parte o la totalidad de Amazon CodeCatalyst no está disponible

Problema: he accedido a la CodeCatalyst consola o he seguido un enlace a ella, pero aparece un error.

Posibles soluciones: los motivos más comunes de este problema son que has seguido un enlace a un proyecto o un espacio al que no te han invitado, o que hay un problema de disponibilidad general con el servicio. Consulta el [informe de Salud](#) para ver si hay algún problema conocido con el servicio. Si no es así, ponte en contacto con la persona que te invitó al proyecto o al espacio y pídele otra invitación. Si no te han invitado a ningún proyecto o espacio, puedes inscribirte y [crear tu propio espacio y tus propios proyectos](#).

No puedo crear un proyecto en CodeCatalyst

Problema: quiero crear un proyecto, pero el botón Crear proyecto aparece como no disponible o recibo un mensaje de error.

Posibles soluciones: los motivos más comunes de este problema son que has iniciado sesión en la consola con un AWS Builder ID que no tiene la función de administrador de Space. Debe tener este rol para crear proyectos en un espacio.

Si tiene este rol y el botón no aparece como disponible, es posible que se trate de un problema transitorio con el servicio. Actualiza el navegador e inténtalo de nuevo.

Solución de problemas de soporte

Aparece un error cuando accedo AWS Support a Amazon CodeCatalyst

Problema: Cuando elijo la CodeCatalyst opción AWS Support para Amazon, recibo el siguiente mensaje de error:

Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Cuenta de AWS that is the billing account for the space.

Posibles soluciones: añada el rol necesario a la cuenta de facturación del espacio.

Cuenta de AWS La cuenta designada como cuenta de facturación del espacio usa el `AWSRoleForCodeCatalystSupport` rol y la política `AmazonCodeCatalystSupportAccess` administrada. Para obtener más información, consulte [Crear el AWSRoleForCodeCatalystSupportrol para tu cuenta y tu espacio](#).

Note

Un ID de AWS constructor solo puede recibir soporte para el alias con el que está autenticado y solo para los recursos en función de los permisos en CodeCatalyst los que esté autenticado. El soporte de cuentas y facturación está disponible para todos los usuarios del espacio. Sin embargo, los creadores solo pueden obtener asistencia para los recursos y la información para los que tengan permisos CodeCatalyst.

No puedo crear casos de soporte técnico para mi espacio

Problema: no puedo crear casos de soporte técnico para mi espacio.

Correcciones: Es necesario añadir un plan Business Support o Enterprise Support a la cuenta de facturación del espacio para que los usuarios del espacio puedan crear casos de soporte técnico. Pide al administrador del espacio que añada un AWS Support plan a tu cuenta de facturación de espacios o visita <https://repost.aws/> para preguntar a la AWS comunidad.

Mi cuenta para casos de asistencia ya no está conectada a mi espacio en CodeCatalyst

Problema: mi cuenta para casos de soporte ya no está conectada a mi espacio en CodeCatalyst.

Soluciones: si un usuario con el rol de administrador del espacio cambia de cuenta de facturación del espacio, el AWS Support plan y todos los casos asociados se desconectarán del espacio. Los AWS Support casos asociados a la antigua cuenta de facturación de espacios dejarán de estar visibles en AWS Support Amazon CodeCatalyst. El usuario raíz de esa cuenta de facturación puede ver y resolver casos antiguos AWS Management Console y configurar permisos de IAM AWS Support para que otros usuarios puedan ver y resolver casos antiguos. No podrá seguir recibiendo asistencia técnica CodeCatalyst desde la antigua cuenta de facturación de Space a través de laAWS Management Console, pero podrá recibir asistencia técnica para otros servicios hasta que se cancele su AWS Support plan.

Para obtener más información, consulte [Actualización, resolución y reapertura de un caso](#) en la Guía del AWS Support usuario.

No puedo abrir un caso de soporte para otro Servicio de AWS en AWS Support Amazon CodeCatalyst

Problema: no puedo abrir un caso de soporte AWS Support para otro Servicio de AWS CodeCatalyst.

Soluciones posibles: solo puedes abrir casos de CodeCatalyst soporte desde AWS Support para CodeCatalyst. Si necesitas soporte para los servicios o recursos desplegados desde CodeCatalyst otro servicio AWS, Amazon u otro servicio de terceros, tendrás que crear un caso a través del canal de soporte del servicio AWS Management Console o del tercero. Para obtener más información, consulta la [sección Creación de casos de soporte y gestión de casos](#) en la Guía del AWS Support usuario.

Parte o la totalidad de Amazon CodeCatalyst no está disponible

Problema: he accedido a la CodeCatalyst consola o he seguido un enlace a ella, pero aparece un error.

Posibles soluciones: los motivos más comunes de este problema son que has seguido un enlace a un proyecto o un espacio al que no te han invitado, o que hay un problema de disponibilidad general con el servicio. Consulta el [informe de Salud](#) para ver si hay algún problema conocido con el servicio. Si no es así, ponte en contacto con la persona que te invitó al proyecto o al espacio y pídele otra invitación. Si no te han invitado a ningún proyecto o espacio, puedes inscribirte y [crear tu propio espacio y tus propios proyectos](#).

No puedo crear un proyecto en CodeCatalyst

Problema: quiero crear un proyecto, pero el botón Crear proyecto aparece como no disponible o recibo un mensaje de error.

Posibles soluciones: los motivos más comunes de este problema son que has iniciado sesión en la consola con un AWS Builder ID que no tiene la función de administrador de Space. Debe tener este rol para crear proyectos en un espacio.

Si tiene este rol y el botón no aparece como disponible, es posible que se trate de un problema transitorio con el servicio. Actualiza el navegador e inténtalo de nuevo.

Quiero enviar comentarios en CodeCatalyst

Problema: he encontrado un error CodeCatalyst y quiero enviar comentarios.

Posibles soluciones: puedes enviar tus comentarios directamente en CodeCatalyst.

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. En el panel de navegación, selecciona Enviar comentarios.
3. Elige el tipo de comentario en el menú desplegable e introduce tu comentario.

Solución de problemas con los repositorios de código fuente

La siguiente información puede ayudarte a solucionar problemas comunes relacionados con los repositorios de código fuente. CodeCatalyst

Temas

- [He alcanzado el límite máximo de almacenamiento de mi espacio y veo advertencias o errores](#)
- [Recibo un error al intentar clonar o enviar archivos a un repositorio de CodeCatalyst código fuente de Amazon](#)
- [Recibo un error al intentar confirmar o enviar información a un repositorio de CodeCatalyst código fuente de Amazon](#)
- [Necesito un repositorio de código fuente para mi proyecto](#)
- [Mi repositorio de código fuente es completamente nuevo, pero contiene una confirmación](#)
- [Quiero una rama diferente como mi rama predeterminada](#)
- [Estoy recibiendo correos electrónicos sobre la actividad de las solicitudes de incorporación de cambios](#)
- [He olvidado mi token de acceso personal \(PAT\)](#)
- [Una solicitud de cambios no muestra los cambios esperados](#)
- [Una solicitud de extracción muestra el estado de No se puede fusionar](#)

He alcanzado el límite máximo de almacenamiento de mi espacio y veo advertencias o errores

Problema: quiero enviar código a uno o más repositorios de código fuente CodeCatalyst, pero aparece un error. En la consola, aparece un mensaje en la página del repositorio de origen que indica que he alcanzado el límite de almacenamiento del espacio.

Soluciones posibles: en función del rol que desempeñes en el proyecto o en el espacio, puedes reducir el tamaño de uno o más de tus repositorios de origen, eliminar los repositorios de origen que no utilices o cambiar tu nivel de facturación por uno que tenga más espacio de almacenamiento.

- Para reducir el tamaño del repositorio de origen de un proyecto, puedes eliminar las ramas que no utilices. Para obtener más información, consulte [Para eliminar una rama \(consola\)](#) y [Función de colaborador](#).
- Para reducir el almacenamiento total de un espacio, puedes eliminar los repositorios de origen no utilizados. Para obtener más información, consulte [Eliminar un repositorio de fuentes](#) y [Función de administrador del proyecto](#).
- Para aumentar la cantidad de almacenamiento disponible para tu espacio, puedes cambiar tu nivel de facturación por uno con más espacio de almacenamiento. Para obtener más información, consulta [Cómo cambiar el nivel CodeCatalyst de facturación](#) en la Guía del CodeCatalyst administrador de Amazon.

Recibo un error al intentar clonar o enviar archivos a un repositorio de CodeCatalyst código fuente de Amazon

Problema: Cuando intento clonar un repositorio de código fuente en un ordenador local o en un entorno de desarrollo integrado (IDE), aparece un error de permisos.

Posibles soluciones: es posible que no tengas un token de acceso personal (PAT) para tu ID de AWS creador, que no hayas configurado tu sistema de administración de credenciales con tu PAT o que tu PAT haya caducado. Pruebe una o más de las siguientes soluciones:

- Cree un token de acceso personal (PAT). Para obtener más información, consulte [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#).
- Asegúrese de haber aceptado una invitación al proyecto que contiene el repositorio de origen y de que sigue siendo miembro de ese proyecto. No puedes clonar un repositorio de código fuente si no eres un miembro activo de ese proyecto. Inicie sesión en la consola e intente navegar hasta

el espacio y el proyecto en los que está intentando clonar un repositorio de código fuente. Si no puedes ver el proyecto en la lista de proyectos del espacio, significa que no eres miembro de ese proyecto o que no has aceptado una invitación a ese proyecto. Para obtener más información, consulte [Aceptar una invitación y crear tu ID de AWS constructor](#).

- Asegúrese de que el comando de clonación tenga el formato correcto e incluya su ID de AWS constructor. Por ejemplo:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

- Úselo AWS CLI para asegurarse de que tiene una PAT asociada a su ID de AWS constructor y de que no ha caducado. Si no tiene una o la PAT ha caducado, cree una. Para obtener más información, consulte [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#).
- Intente crear un entorno de desarrollo para trabajar con el código del repositorio fuente en lugar de clonarlo en un repositorio o IDE local. Para obtener más información, consulte [Creación de un entorno de desarrollo](#).

Recibo un error al intentar confirmar o enviar información a un repositorio de CodeCatalyst código fuente de Amazon

Problema: cuando intento subirlo a un repositorio de origen, aparece un error de permisos.

Posibles soluciones: es posible que no tengas un rol en el proyecto que te permita confirmar e insertar los cambios de código en el proyecto. Consulta tu función en el proyecto en el que estás intentando insertar los cambios en un repositorio fuente. Para obtener más información, consulte [Ver los miembros de un proyecto](#) y [Trabajar con roles en Amazon CodeCatalyst](#).

Si tienes un rol que te permite confirmar e insertar cambios, es posible que la rama en la que intentes confirmar los cambios tenga configurada una regla de ramificación que te impida enviar cambios de código a esa rama. Intenta crear una rama y, en su lugar, enviar el código a esa rama. Para obtener más información, consulte [Gestione las acciones permitidas para una sucursal con las reglas de la sucursal](#).

Necesito un repositorio de código fuente para mi proyecto

Problema: mi proyecto no tiene un repositorio de código fuente o necesito otro repositorio de código fuente para mi proyecto.

Posibles soluciones: algunos proyectos se crean sin recursos. Si eres miembro del proyecto, puedes crear repositorios de código fuente para ese proyecto en CodeCatalyst. Si alguien con el rol de administrador del espacio instala los GitHub repositorios y los conecta a una GitHub cuenta, puedes vincular los GitHub repositorios disponibles para añadirlos a tu proyecto si tienes el rol de administrador del proyecto. Para obtener más información, consulte [Crear un repositorio de origen y Vincular un repositorio](#) de origen.

Mi repositorio de código fuente es completamente nuevo, pero contiene una confirmación

Problema: acabo de crear un repositorio de código fuente. Debería estar vacío, pero contiene una confirmación, una rama y un README .md archivo.

Posibles soluciones: este es el comportamiento esperado. Todos los repositorios de código fuente CodeCatalyst incluyen una confirmación inicial que establece la rama predeterminada en un código de muestra (si el repositorio se creó para un proyecto con un esquema que incluía un código de ejemplo) o un archivo de plantilla de Markdown para un archivo README del repositorio. `main` Puedes crear ramas adicionales en la consola y en los clientes de Git. Puede crear y editar archivos en la consola y eliminar archivos en entornos de desarrollo y clientes Git.

Quiero una rama diferente como mi rama predeterminada

Problema: mi repositorio de origen venía con una rama predeterminada con un nombre `main`, pero quiero una rama diferente como rama predeterminada.

Posibles soluciones: no puedes cambiar ni eliminar la rama predeterminada en los repositorios de origen de CodeCatalyst. Puede crear ramas adicionales y utilizarlas en las acciones de origen de los flujos de trabajo. También puedes optar por vincular GitHub repositorios y usarlos como repositorios para tu proyecto.

Estoy recibiendo correos electrónicos sobre la actividad de las solicitudes de incorporación de cambios

Problema: no me registré ni configuré las notificaciones por correo electrónico sobre la actividad de las solicitudes de extracción, pero las recibo de todos modos.

Posibles soluciones: las notificaciones por correo electrónico sobre la actividad de las solicitudes de extracción se envían automáticamente. Para obtener más información, consulte [Trabajar con solicitudes de cambios en Amazon CodeCatalyst](#).

He olvidado mi token de acceso personal (PAT)

Problema: he estado usando una PAT para clonar, insertar y extraer código de los repositorios fuente, pero he perdido el valor de mi token y no lo encuentro en la CodeCatalyst consola.

Posibles soluciones: la forma más rápida de resolver este problema es crear otra PAT y configurar el administrador de credenciales o el IDE para que utilice esta nueva PAT. Solo mostramos el valor de una PAT cuando la creas. Si pierde este valor, no podrá recuperarlo. Para obtener más información, consulte [Gestión de los tokens de acceso personal en Amazon CodeCatalyst](#).

Una solicitud de cambios no muestra los cambios esperados

Problema: he creado una solicitud de incorporación de cambios, pero no veo los cambios que esperaba ver entre las sucursales de origen y destino.

Posibles soluciones: esto puede deberse a varios problemas. Pruebe una o más de las siguientes soluciones:

- Es posible que esté revisando los cambios entre revisiones anteriores o que no esté viendo los cambios más recientes. Actualiza tu navegador y asegúrate de haber elegido la comparación entre las revisiones que desees ver.
- No todos los cambios de una solicitud de cambios se pueden mostrar en la consola. Por ejemplo, no puedes ver los submódulos de Git en la consola, por lo que no puedes ver las diferencias de un submódulo en una solicitud de cambios. Es posible que algunas diferencias sean demasiado grandes para mostrarlas. Para obtener más información, consulte [Cuotas para los repositorios de fuentes en CodeCatalyst](#) y [Visualización de un archivo](#).
- Las solicitudes de extracción muestran las diferencias entre la base de fusión y la revisión que elija. Cuando creas una solicitud de extracción, la diferencia que se muestra es la diferencia entre la punta de la rama de origen y la punta de la rama de destino. Una vez creada la solicitud de extracción, la diferencia que se muestra es entre la revisión y su base de fusión. La base de fusión es la confirmación que estaba en la punta de la rama de destino cuando se creó la revisión. La base de fusión puede cambiar entre revisiones. Para obtener más información sobre las diferencias y las bases de fusión en Git, consulta [git-merge-base](#) la documentación de Git.

Una solicitud de extracción muestra el estado de No se puede fusionar

Problema: quiero fusionar una solicitud de extracción, pero su estado es No fusionable.

Posibles soluciones: esto puede deberse a uno o más problemas:

- Todos los revisores necesarios para tu solicitud de extracción deben aprobarla antes de que se pueda fusionar. Revisa la lista de revisores obligatorios para ver si hay algún revisor con un icono de reloj junto al nombre. El icono de un reloj indica que el revisor no ha aprobado la solicitud de revisión.

Note

Si un revisor obligatorio ha sido eliminado de tu proyecto antes de aprobar la solicitud de extracción, no podrás fusionar la solicitud de extracción. Cierra la solicitud de extracción y crea una nueva.

- Es posible que haya un conflicto de fusión entre la rama de origen y la de destino. CodeCatalyst no es compatible con todas las posibles estrategias y opciones de fusión de Git. Puede evaluar las ramas para detectar conflictos de fusión en un entorno de desarrollo o clonar el repositorio y utilizar un IDE o las herramientas de Git para buscar y resolver los conflictos de fusión. Para obtener más información, consulte [Fusión de una solicitud de extracción](#).

Solución de problemas de proyectos y planos

Esta sección puede ayudarte a solucionar algunos problemas comunes que puedes encontrar al trabajar con proyectos y planos en Amazon. CodeCatalyst

A la API de Java le faltan dependencias en el AWS Fargate blueprint para apache-maven-3.8.6

Problema: en el caso de un proyecto creado a partir de la API de Java con un AWS Fargate blueprint, el flujo de trabajo falla y se produce un error debido a la falta de dependencias. apache-maven-3.8.6 El flujo de trabajo produce un error con un resultado similar al del siguiente ejemplo:

```
Step 8/25 : RUN wget https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/apache-
maven-3.8.6-bin.tar.gz -P /tmp
---> Running in 1851ce6f4d1b
[91m--2023-03-10 01:24:55-- https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving d1cdn.apache.org (d1cdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.apache.org)|151.101.2.132|:443...
```

```
[0m[91mconnected.  
[0m[91mHTTP request sent, awaiting response... [0m[91m404 Not Found  
2023-03-10 01:24:55 ERROR 404: Not Found.  
[0mThe command '/bin/sh -c wget https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/  
apache-maven-3.8.6-bin.tar.gz -P /tmp' returned a non-zero code: 8  
[Container] 2023/03/10 01:24:55 Command failed with exit status 8
```

Solución: actualice el Dockerfile del blueprint mediante los siguientes pasos.

1. En la barra de búsqueda, ingresa `apache-maven-3.8.6` para ubicar el dockerfile dentro del proyecto creado con la API de Java con blueprint. AWS Fargate
2. Actualiza el Dockerfile (`/static-assets/app/Dockerfile`) para usarlo `maven:3.9.0-amazoncorretto-11` como imagen base y elimina la dependencia del paquete. `apache-maven-3.8.6`
3. (Recomendado) También recomendamos actualizar el tamaño del montón de Maven a 6 GB.

A continuación se muestra un ejemplo de Dockerfile.

```
FROM maven:3.9.0-amazoncorretto-11 AS builder  
  
COPY ./pom.xml ./pom.xml  
COPY src ./src/  
  
ENV MAVEN_OPTS='-Xmx6g'  
  
RUN mvn -Dmaven.test.skip=true clean package  
  
FROM amazoncorretto:11-alpine  
  
COPY --from=builder target/CustomerService-0.0.1.jar CustomerService-0.0.1.jar  
EXPOSE 80  
CMD ["java", "-jar", "-Dspring.profiles.active=prod", "/CustomerService-0.0.1.jar", "-server.port=80"]
```

El flujo de trabajo moderno del blueprint de aplicaciones web de tres niveles OnPullRequestfalla debido a un error de permisos para Amazon CodeGuru

Problema: cuando intento ejecutar un flujo de trabajo para mi proyecto, el flujo de trabajo no se ejecuta y aparece el siguiente mensaje:

```
Failed at codeguru_codereview: The action failed during runtime. View the action's logs for more details.
```

Solución: una posible causa del error de esta acción podría deberse a la falta de permisos en la política de funciones de IAM, ya que a su versión de la función de servicio utilizada por CodeCatalyst la persona conectada le faltan los permisos necesarios para que la acción `codeguru_codereview` Cuenta de AWS se ejecute correctamente. Para solucionar este problema, el rol de servicio debe actualizarse con los permisos necesarios o debe cambiar el rol de servicio utilizado para el flujo de trabajo por uno que tenga los permisos necesarios para Amazon CodeGuru y Amazon CodeGuru Reviewer. Siga estos pasos para encontrar su función y actualizar los permisos de la política de funciones para permitir que el flujo de trabajo se ejecute correctamente.

Note

Estos pasos se aplican a los siguientes flujos de trabajo en CodeCatalyst:

- El `OnPullRequest` flujo de trabajo previsto para los proyectos creados con el modelo de aplicación web moderna de tres niveles en CodeCatalyst
- Flujos de trabajo agregados a los proyectos CodeCatalyst con acciones que acceden a Amazon CodeGuru o Amazon CodeGuru Reviewer.

Cada proyecto contiene flujos de trabajo con acciones que utilizan un rol y un entorno proporcionados por la persona Cuenta de AWS conectada al proyecto CodeCatalyst. El flujo de trabajo con las acciones y su política designada se almacena en el repositorio de origen, en el directorio `/.codecatalyst/workflows`. No es necesario modificar el flujo de trabajo (YAML), a menos que añada un nuevo identificador de rol al flujo de trabajo existente. Para obtener información sobre los elementos y el formato de la plantilla YAML, consulte [Referencia de definición de flujo de trabajo](#)

Estos son los pasos básicos que debes seguir para editar tu política de roles y verificar el flujo de trabajo YAML.

Para hacer referencia al nombre de tu rol en el flujo de trabajo (YAML) y actualizar la política

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio. Navega hasta tu proyecto.
3. Elija CI/CD y, a continuación, elija Flujos de trabajo.

4. Elija el flujo de trabajo titulado. OnPullRequest Elija la pestaña Definition (Definición).
5. En el flujo de trabajo YAML, en el Role: campo situado debajo de la acción codeguru_codereview, anota el nombre del rol. Este es el rol con la política que modificará en IAM. En el siguiente ejemplo se muestra el nombre del rol.

The screenshot displays the Amazon CodeCatalyst interface for configuring a workflow. The top navigation bar includes 'CI/CD', 'Workflows', 'Environments', 'Compute', 'Secrets', and 'Change tracking'. The main header shows the workflow name 'OnPullRequest' for repository 'mysfitscvb63' and test 'test'. Below the header, there are tabs for 'Latest state', 'Definition', and 'Runs'. A summary table provides details on the latest run, run mode, latest commit, and workflow definition status. The 'Definition' tab is selected, showing a visual diagram of the workflow and a detailed YAML configuration. The diagram illustrates a 'WorkflowSource' connected to a 'codeguru_codereview' action. The YAML configuration specifies triggers for 'PULLREQUEST' on the 'main' branch and an action named 'codeguru_codereview' with an identifier 'aws/build@v1'. The role specified in the configuration is 'CodeCatalystPreviewDevelopmentAdministrator-j'.

6. Haga una de las siguientes acciones:
 - (Recomendado) Actualice el rol de servicio conectado a su proyecto con los permisos necesarios para Amazon CodeGuru y Amazon CodeGuru Reviewer. El rol tendrá un nombre `CodeCatalystWorkflowDevelopmentRole-spaceName` con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#). Continúe con los siguientes pasos para actualizar la política en IAM.

Note

Debe tener acceso de AWS administrador a la Cuenta de AWS función y la política.

- Cambie el rol de servicio utilizado para el flujo de trabajo por uno que tenga los permisos necesarios para Amazon CodeGuru y Amazon CodeGuru Reviewer o cree un nuevo rol con los permisos necesarios.
7. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

En la consola de IAM, busque el rol del paso 5, por ejemploCodeCatalystPreviewDevelopmentRole.

8. En el rol del paso 5, cambie la política de permisos para incluir los `codeguru:*` permisos `codeguru-reviewer:*` y. Tras añadir estos permisos, la política de permisos debería tener un aspecto similar al siguiente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "apigateway:*",
        "ecr:*",
        "ecs:*",
        "ssm:*",
        "codedeploy:*",
        "s3:*",
        "iam:DeleteRole",
        "iam:UpdateRole",
        "iam:Get*",
        "iam:TagRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:CreatePolicy",
        "iam>DeletePolicy",
```

```

        "iam:CreatePolicyVersion",
        "iam:DeletePolicyVersion",
        "iam:PutRolePermissionsBoundary",
        "iam:DeleteRolePermissionsBoundary",
        "sts:AssumeRole",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:DescribeRules",
        "elasticloadbalancing:ModifyRule",
        "cloudwatch:DescribeAlarms",
        "sns:Publish",
        "sns:ListTopics",
        "codeguru-reviewer:*",
        "codeguru:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
}

```

- Tras realizar las correcciones de política, vuelva a la ejecución del flujo de trabajo CodeCatalyst y vuelva a iniciarla.

¿Sigues intentando resolver tu problema?

Puedes ir a [Amazon CodeCatalyst](#) o rellenar un formulario de [comentarios de Support](#). En la sección Solicitud de información, en Cómo podemos ayudarte, indica que eres CodeCatalyst cliente de Amazon. Proporciona tantos detalles como sea posible para que podamos abordar tu problema de la manera más eficiente.

Solución de problemas con los flujos de trabajo

Consulta las siguientes secciones para solucionar problemas relacionados con los flujos de trabajo en Amazon CodeCatalyst. Para obtener más información acerca de los flujos de trabajo, consulte [???](#).

Temas

- [¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?](#)

- [¿Cómo puedo corregir los errores «La definición de flujo de trabajo no tiene errores»?](#)
- [¿Cómo puedo corregir los errores «No se pueden localizar las credenciales» y ExpiredToken «»?](#)
- [¿Cómo puedo corregir los errores «No se puede conectar al servidor»?](#)
- [¿Por qué faltan CodeDeploy campos en el editor visual?](#)
- [¿Cómo puedo corregir los errores de las capacidades de IAM?](#)
- [¿Cómo puedo corregir los errores de «npm install»?](#)
- [¿Por qué varios flujos de trabajo tienen el mismo nombre?](#)
- [¿Puedo almacenar mis archivos de definición de flujo de trabajo en otra carpeta?](#)
- [¿Cómo puedo añadir acciones secuencialmente a mi flujo de trabajo?](#)
- [¿Por qué mi flujo de trabajo se valida correctamente pero falla en tiempo de ejecución?](#)
- [La detección automática no detecta ningún informe sobre mi acción](#)
- [Mi acción falla en los informes detectados automáticamente después de configurar los criterios de éxito](#)
- [La detección automática genera informes que no deseo](#)
- [La detección automática genera muchos informes pequeños para un único marco de pruebas](#)
- [Los flujos de trabajo enumerados en CI/CD no coinciden con los del repositorio de origen](#)
- [No puedo crear ni actualizar flujos de trabajo](#)

¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?

Problema: en la CodeCatalyst consola, en CI/CD, Flujos de trabajo, su flujo de trabajo aparece con el siguiente mensaje:

```
Workflow is inactive.
```

Este mensaje indica que el archivo de definición del flujo de trabajo contiene un activador que no se aplica a la rama en la que se encuentra actualmente. Por ejemplo, el archivo de definición de flujo de trabajo puede contener un PUSH disparador que haga referencia a su main rama, pero usted se encuentra en una rama de características. Como los cambios que realizas en la rama de entidades no se aplican a main la rama de funciones ni iniciarán las ejecuciones en ellamain, retira CodeCatalyst el flujo de trabajo de la rama y lo marca como talInactive.

Soluciones posibles:

Si quieres iniciar un flujo de trabajo en tu rama de funciones, puedes hacer lo siguiente:

- En la rama de entidades, en el archivo de definición del flujo de trabajo, elimina la `Branches` propiedad de la `Triggers` sección para que tenga el siguiente aspecto:

```
Triggers:  
- Type: PUSH
```

Esta configuración hace que el activador se active cuando se envía a cualquier rama, incluida la rama de funciones. Si el activador está activado, CodeCatalyst iniciará la ejecución de un flujo de trabajo con el archivo de definición del flujo de trabajo y los archivos fuente de la rama a la que esté accediendo.

- En la rama de funciones, en el archivo de definición del flujo de trabajo, elimina la `Triggers` sección y ejecuta el flujo de trabajo manualmente.
- En la rama de entidades, en el archivo de definición del flujo de trabajo, cambia la `PUSH` sección para que haga referencia a la rama de entidades en lugar de a otra rama (por ejemplo). `main`

Important

Tenga cuidado de no realizar estos cambios si no tiene intención de fusionarlos para volver a formar parte de su `main` rama.

Para obtener más información sobre la edición del archivo de definición del flujo de trabajo, consulte [Para editar un flujo de trabajo](#).

Para obtener más información acerca de los disparadores, consulte [Trabajar con factores desencadenantes](#).

¿Cómo puedo corregir los errores «La definición de flujo de trabajo **no** tiene errores»?

Problema: aparece alguno de los siguientes mensajes de error:

Error 1:

En la página de flujos de trabajo de CI/CD, debajo del nombre de su flujo de trabajo, verá:

Workflow definition has *n* errors

Error 2:

Al editar un flujo de trabajo, se pulsa el botón Validar y aparece el siguiente mensaje en la parte superior de la CodeCatalyst consola:

```
The workflow definition has errors. Fix the errors and choose Validate to verify your changes.
```

Error 3:

Tras acceder a la página de detalles del flujo de trabajo, aparece el siguiente error en el campo de definición del flujo de trabajo:

```
n errors
```

Soluciones posibles:

- Elija CI/CD, elija Flujos de trabajo y elija el nombre del flujo de trabajo que contiene el error. En el campo de definición del flujo de trabajo, situado cerca de la parte superior, selecciona el enlace al error. Los detalles sobre el error aparecen en la parte inferior de la página. Sigue los consejos de solución de problemas que aparecen en el error para solucionar el problema.
- Asegúrese de que el archivo de definición del flujo de trabajo sea un archivo YAML.
- Asegúrese de que las propiedades YAML del archivo de definición del flujo de trabajo estén anidadas en el nivel correcto. Para ver cómo deben anidarse las propiedades en el archivo de definición del flujo de trabajo, consulte la documentación de la [Referencia de definición de flujo de trabajo](#) acción o consulte la misma, a la que se puede acceder desde. [Añadir una acción](#)
- Asegúrese de que los asteriscos (*) y otros caracteres especiales estén bien alineados. Para evitarlos, añada comillas simples o dobles. Por ejemplo:

```
Outputs:  
  Artifacts:  
    - Name: myartifact  
      Files:  
        - "/**/*"
```

Para obtener más información sobre los caracteres especiales del archivo de definición del flujo de trabajo, consulte [Pautas y convenciones de sintaxis](#).

- Asegúrese de que las propiedades YAML del archivo de definición del flujo de trabajo utilicen las mayúsculas correctas. Para obtener más información sobre las reglas de uso de mayúsculas, consulte [Pautas y convenciones de sintaxis](#). Para determinar la carcasa correcta de cada propiedad, consulte o consulte la [Referencia de definición de flujo de trabajo](#) documentación de su acción, que está vinculada a [Añadir una acción](#).
- Asegúrese de que la `SchemaVersion` propiedad esté presente y configurada en la versión correcta en el archivo de definición del flujo de trabajo. Para obtener más información, consulte [SchemaVersion](#).
- Asegúrese de que la `Triggers` sección del archivo de definición del flujo de trabajo incluya todas las propiedades necesarias. Para determinar las propiedades requeridas, elija el activador en el [editor visual](#) y busque los campos en los que falte información, o consulte la documentación de referencia del activador en [Triggers](#).
- Asegúrese de que la `DependsOn` propiedad del archivo de definición del flujo de trabajo esté configurada correctamente y no introduzca dependencias circulares. Para obtener más información, consulte [Configurar acciones para que dependan de otras acciones](#).
- Asegúrese de que la `Actions` sección del archivo de definición del flujo de trabajo incluya al menos una acción. Para obtener más información, consulte [Acciones](#).
- Asegúrese de que cada acción incluya todas las propiedades obligatorias. Para determinar las propiedades obligatorias, elija la acción en el [editor visual](#) y busque los campos en los que falte información, o consulte la documentación de la acción, que está vinculada desde [Añadir una acción](#).
- Asegúrese de que todos los artefactos de entrada tengan los artefactos de salida correspondientes. Para obtener más información, consulte [Definir un artefacto de salida](#).
- Asegúrese de que las variables definidas en una acción se exporten para poder utilizarlas en otras acciones. Para obtener más información, consulte [Exportar una variable para que otras acciones puedan utilizarla](#).

¿Cómo puedo corregir los errores «No se pueden localizar las credenciales» y `ExpiredToken` «»?

Problema: Mientras lo solucionas [Tutorial: Implementación de una aplicación en Amazon EKS](#), ves uno o ambos de los siguientes mensajes de error en la ventana del terminal de tu máquina de desarrollo:

Unable to locate credentials. You can configure credentials by running "aws configure".


ExpiredToken: The security token included in the request is expired

Soluciones posibles:

Estos errores indican que las credenciales que utilizas para acceder a AWS los servicios han caducado. En este caso, no ejecute el `aws configure` comando. En su lugar, siga las instrucciones siguientes para actualizar la clave de AWS acceso y el token de sesión.

Para actualizar la clave de AWS acceso y el token de sesión

1. Asegúrese de tener la URL del portal de AWS acceso, el nombre de usuario y la contraseña del usuario que está utilizando en el tutorial completo de Amazon EKS (`codecatalyst-eks-user`). Debería haber configurado estos elementos cuando completó [Paso 1: Configura tu máquina de desarrollo](#) el tutorial.

 Note

Si no dispone de esta información, vaya a la página de `codecatalyst-eks-user` detalles del Centro de Identidad de IAM, seleccione Restablecer la contraseña, Generar una contraseña de un solo uso [...] y vuelva a restablecer la contraseña para que aparezca la información en la pantalla.

2. Realice una de las acciones siguientes:
 - Pegue la URL del portal de AWS acceso en la barra de direcciones de su navegador.
Or (Disyunción)
 - Actualice la página del portal de AWS acceso si ya está cargada.
3. Inicie sesión con el nombre `codecatalyst-eks-user` de usuario y la contraseña, si aún no lo ha hecho.
4. Elija y Cuenta de AWS, a continuación, elija el nombre del usuario Cuenta de AWS al que ha asignado el conjunto de permisos y el `codecatalyst-eks-user` usuario.
5. Junto al nombre del conjunto de permisos (`codecatalyst-eks-permission-set`), elija Línea de comandos o Acceso mediante programación.
6. Copie los comandos que se encuentran en el centro de la página. Tienen un aspecto similar al siguiente:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... donde *session-token* es una cadena aleatoria larga.

7. Pegue los comandos en el indicador del terminal de su máquina de desarrollo y presione Entrar.

Se cargan las nuevas claves y el token de sesión.

Ya ha actualizado sus credenciales. Los `kubectl` comandos `AWS CLI` `eksctl`, y deberían funcionar ahora.

¿Cómo puedo corregir los errores «No se puede conectar al servidor»?

Problema: Mientras trabaja en el tutorial descrito en [Tutorial: Implementación de una aplicación en Amazon EKS](#), verá un mensaje de error similar al siguiente en la ventana del terminal de su máquina de desarrollo:

```
Unable to connect to the server: dial tcp: lookup long-string.gr7.us-west-2.eks.amazonaws.com on 1.2.3.4:5: no such host
```

Soluciones posibles:

Este error suele indicar que las credenciales que utiliza la `kubectl` utilidad para conectarse al clúster de Amazon EKS han caducado. Para resolver el problema, actualice las credenciales introduciendo el siguiente comando en la línea de comandos del terminal:

```
aws eks update-kubeconfig --name codecatalyst-eks-cluster --region us-west-2
```

Donde:

- *codecatalyst-eks-cluster* se sustituye por el nombre de su clúster de Amazon EKS.
- *us-west-2* se sustituye por la región en la que está AWS desplegado el clúster.

¿Por qué faltan CodeDeploy campos en el editor visual?

Problema: está utilizando una acción de [implementación en Amazon ECS](#) y no ve los CodeDeploy campos, como CodeDeploy AppSpec en el editor visual del flujo de trabajo. Este problema puede

deberse a que el servicio Amazon ECS que especificó en el campo Servicio no está configurado para realizar despliegues en azul o verde.

Soluciones posibles:

- Elija un servicio de Amazon ECS diferente en la pestaña Configuración de la acción Implementar en Amazon ECS. Para obtener más información, consulte [Añadir la acción «Implementar en Amazon ECS»](#).
- Configure el servicio Amazon ECS seleccionado para realizar despliegues azules o verdes. Para obtener más información sobre la configuración de las implementaciones azul/verde, consulte Implementación [azul/verde en CodeDeploy](#) la Guía para desarrolladores de Amazon Elastic Container Service.

¿Cómo puedo corregir los errores de las capacidades de IAM?

Problema: estás utilizando una acción de [despliegue de AWS CloudFormation pila](#) y la ves `##[error] requires capabilities: [capability-name]` en los registros de la acción de despliegue de AWS CloudFormation pila.

Posibles soluciones: complete el siguiente procedimiento para añadir la capacidad al archivo de definición del flujo de trabajo. Para obtener más información sobre las capacidades de IAM, consulte [Reconocimiento de los recursos de IAM en AWS CloudFormation las plantillas](#) de la Guía del usuario de IAM.

Visual

Para añadir una capacidad de IAM mediante el editor visual

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elija Visual.
7. En el diagrama de flujo de trabajo, elige la acción Implementar AWS CloudFormation pila.

8. Elija la pestaña Configuración.
9. En la parte inferior, selecciona Avanzado (opcional).
10. En la lista desplegable de capacidades, seleccione la casilla de verificación situada junto a la capacidad mencionada en el mensaje de error. Si la capacidad no está disponible en la lista, usa el editor YAML para añadirla.
11. (Opcional) Selecciona Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
12. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.
13. Si la ejecución de un nuevo flujo de trabajo no se inicia automáticamente, ejecútelo manualmente para comprobar si los cambios corrigen el error. Para obtener más información sobre la ejecución manual de un flujo de trabajo, consulte [Iniciar una ejecución de flujo de trabajo](#).

YAML

Para añadir una función de IAM mediante el editor YAML

1. [Abra la CodeCatalyst consola en https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elija el proyecto.
3. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
4. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
5. Elija Editar.
6. Elige YAML.
7. En la acción Implementar AWS CloudFormation pila, añada una `capabilities` propiedad, como esta:

```
DeployCloudFormationStack:  
  Configuration:  
    capabilities: capability-name
```

Sustituya el *nombre de la capacidad de IAM* por el nombre de la capacidad de IAM que aparece en el mensaje de error. Utilice comas y sin espacios para enumerar varias

capacidades. Para obtener más información, consulte la descripción de la `capabilities` propiedad en [Referencia de acción «Implementar AWS CloudFormation pila»](#)

8. (Opcional) Elija Validar para validar el código YAML del flujo de trabajo antes de confirmarlo.
9. Selecciona Confirmar, introduce un mensaje de confirmación y vuelve a seleccionar Confirmar.
10. Si la ejecución de un nuevo flujo de trabajo no se inicia automáticamente, ejecútelo manualmente para comprobar si los cambios corrigen el error. Para obtener más información sobre la ejecución manual de un flujo de trabajo, consulte [Iniciar una ejecución de flujo de trabajo](#).

¿Cómo puedo corregir los errores de «npm install»?

Problema: la acción de [AWS CDK despliegue o la acción](#) de [AWS CDK arranque](#) fallan y se produce un error. `npm install` Este error puede producirse porque estás almacenando las dependencias de tu AWS CDK aplicación en un registro de administrador de paquetes de nodos privados (npm) al que no se puede acceder mediante la acción.

Posibles soluciones: sigue las siguientes instrucciones para actualizar el `cdk.json` archivo de tu AWS CDK aplicación con información adicional de registro y autenticación.

Antes de empezar

1. Crea secretos para tu información de autenticación. Hará referencia a estos secretos en el `cdk.json` archivo en lugar de proporcionar los equivalentes en texto claro. Para crear los secretos:
 - a. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
 - b. Elija el proyecto.
 - c. En el panel de navegación, elija CI/CD y, a continuación, Secretos.
 - d. Cree dos secretos con las siguientes propiedades:

Primer secreto	Segundo secreto
Nombre: <code>npmUsername</code>	Nombre: <code>npmAuthToken</code>
Valor: <i>npm-username</i> , donde <i>npm-username</i> es el nombre de usuario que	Valor: <i>npm-auth-token</i> , donde <i>npm-auth-token</i> se encuentra el token de

Primer secreto	Segundo secreto
<p>se utiliza para autenticarse en su registro npm privado.</p> <p>(Opcional) Descripción: The username used to authenticate to the private npm registry.</p>	<p>acceso que se utiliza para autenticarse en su registro npm privado. Para obtener más información sobre los tokens de acceso de npm, consulta Acerca de los tokens de acceso en la documentación de npm.</p> <p>Descripción (opcional): The access token used to authenticate to the private npm registry.</p>

Para obtener más información sobre los secretos, consulte [Trabajando con secretos](#).

2. Añada los secretos como variables de entorno a su AWS CDK acción. La acción reemplazará las variables por valores reales cuando se ejecute. Para añadir los secretos:
 - a. En el panel de navegación, elija CI/CD y, a continuación, elija Flujos de trabajo.
 - b. Elija el nombre de su flujo de trabajo. Puede filtrar por el nombre del repositorio o la rama de origen donde está definido el flujo de trabajo, o bien filtrar por el nombre del flujo de trabajo.
 - c. Elija Editar.
 - d. Elija Visual.
 - e. En el diagrama de flujo de trabajo, elige tu AWS CDK acción.
 - f. Seleccione la pestaña Entradas.
 - g. Añada dos variables con las siguientes propiedades:

Primera variable	Segunda variable
Nombre: NPMUSER	Nombre: NPMTOKEN
Valor: <code>\${Secrets.npmUsername}</code>	Valor: <code>\${Secrets.npmAuthToken}</code>

Ahora tiene dos variables que contienen referencias a secretos.

El código YAML del archivo de definición de flujo de trabajo debería tener un aspecto similar al siguiente:

Note

El siguiente ejemplo de código proviene de una acción de AWS CDK arranque; una acción de AWS CDK despliegue tendrá un aspecto similar.

```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
  CDKBootstrapAction:
    Identifier: aws/cdk-bootstrap@v1
    Inputs:
      Variables:
        - Name: NPMUSER
          Value: ${Secrets.npmUsername}
        - Name: NPMTOKEN
          Value: ${Secrets.npmAuthToken}
      Sources:
        - WorkflowSource
    Environment:
      Name: Dev2
      Connections:
        - Name: account-connection
          Role: codecatalystAdmin
    Configuration:
      Parameters:
        Region: "us-east-2"
```

Ahora está listo para usar las NPMTOKEN variables NPMUSER y del `cdk.json` archivo. Vaya al siguiente procedimiento.

Para actualizar el archivo `cdk.json`

1. Cambia al directorio raíz de tu AWS CDK proyecto y abre el `cdk.json` archivo.
2. Busque la `"app"`: propiedad y cámbiela para incluir el código que aparece en *cursiva roja*:

Note

El siguiente código de ejemplo proviene de un TypeScript proyecto. Si utilizas un JavaScript proyecto, el código tendrá un aspecto similar, aunque no idéntico.

```
{
  "app": "npm set registry=https://your-registry/folder/CDK-package/ --
  userconfig .npmrc && npm set //your-registry/folder/CDK-package/:always-auth=true
  --userconfig .npmrc && npm set //your-registry/folder/CDK-package/:_authToken=
  \"${NPMUSER}\"\": \"${NPMTOKEN}\" && npm install && npx ts-node --prefer-ts-exts bin/
  hello-cdk.ts|js",
  "watch": {
    "include": [
      "*"
    ],
  },
  "exclude": [
    "README.md",
    "cdk*.json",
    "**/*.d.ts",
    "**/*.js",
    "tsconfig.json",
    "package*.json",
  ],
  ...
}
```

3. En el código resaltado en *cursiva roja*, sustituya:

- *Your-registry/folder/CDK-Package/* por la ruta a las dependencias de su proyecto en su registro privado. AWS CDK
- *hello-cdk.ts|js con el* nombre de tu archivo de punto de entrada. Puede ser un archivo .ts (TypeScript) o .js (JavaScript) según el idioma que utilices.

Note

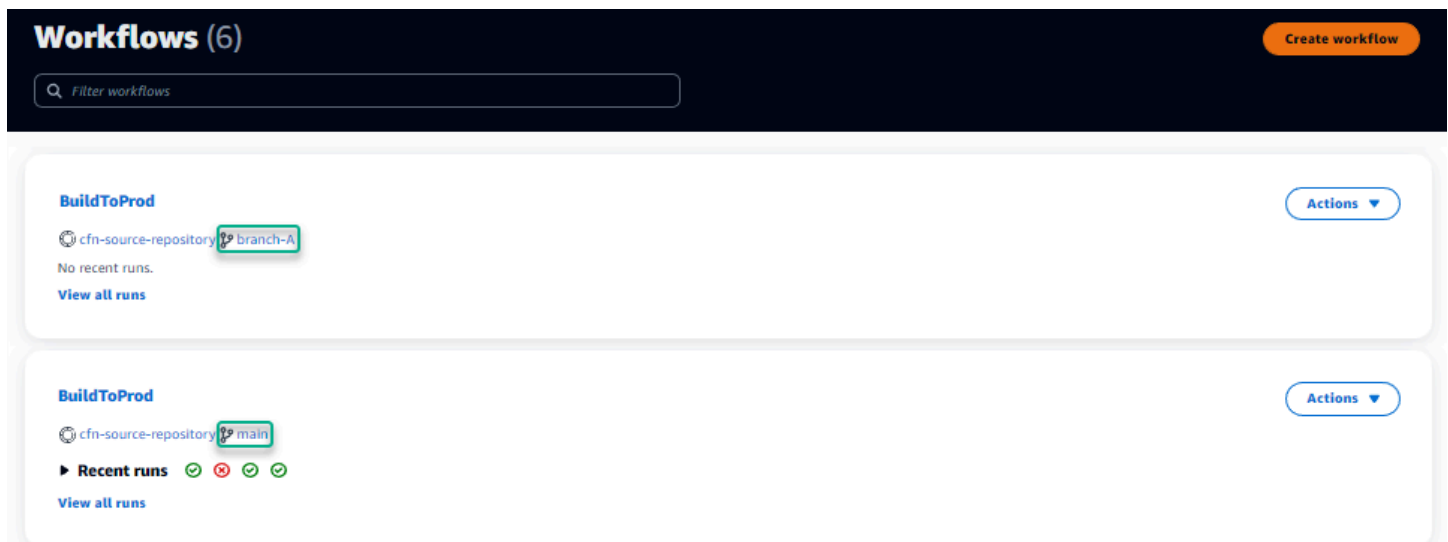
La acción sustituirá las variables NPMUSER y *NPMTOKEN por el nombre de usuario y el token* de acceso de npm que especificaste en Secrets.

4. Guarde el archivo `cdk.json`.

5. Vuelve a ejecutar la acción manualmente para comprobar si los cambios corrigen el error. Para obtener más información sobre la ejecución manual de acciones, consulte [Iniciar una ejecución de flujo de trabajo](#).

¿Por qué varios flujos de trabajo tienen el mismo nombre?

Los flujos de trabajo se almacenan por rama y repositorio. Dos flujos de trabajo diferentes pueden tener el mismo nombre si existen en ramas diferentes. En la página Flujos de trabajo, puede diferenciar los flujos de trabajo del mismo nombre consultando el nombre de la rama. Para obtener más información, consulte [Trabajar con sucursales en Amazon CodeCatalyst](#).



The screenshot shows the 'Workflows (6)' page in Amazon CodeCatalyst. At the top right is a 'Create workflow' button. Below it is a search bar labeled 'Filter workflows'. Two workflow cards are displayed:

- The first card is for 'BuildToProd' in the 'cfn-source-repository' repository, specifically for the 'branch-A' branch. It shows 'No recent runs' and a 'View all runs' link.
- The second card is also for 'BuildToProd' in the 'cfn-source-repository' repository, but for the 'main' branch. It shows 'Recent runs' with four status icons (one green, one red, one green, one green) and a 'View all runs' link.

¿Puedo almacenar mis archivos de definición de flujo de trabajo en otra carpeta?

No, debe almacenar todos los archivos de definición de flujos de trabajo en la `.codecatalyst/workflows` carpeta. Si utiliza un repositorio mono con varios proyectos lógicos, coloque todos los archivos de definición de flujo de trabajo en la `.codecatalyst/workflows` carpeta y, a continuación, utilice la `FilesChanged` propiedad dentro de un disparador para activar el flujo de trabajo en una ruta de proyecto específica. Para obtener más información, consulte [Trabajar con factores desencadenantes](#).

¿Cómo puedo añadir acciones secuencialmente a mi flujo de trabajo?

De forma predeterminada, cuando añadas una acción a tu flujo de trabajo, no tendrá dependencias y se ejecutará en paralelo con otras acciones.

Si desea organizar las acciones en secuencia, puede establecer una dependencia de otra acción configurando el `DependsOn` campo. También puedes configurar una acción para que consuma artefactos o variables que sean el resultado de otras acciones. Para obtener más información, consulte [Configurar acciones para que dependan de otras acciones](#).

¿Por qué mi flujo de trabajo se valida correctamente pero falla en tiempo de ejecución?

Si validó su flujo de trabajo con el `Validate` botón, pero el flujo de trabajo falló de todos modos, podría deberse a una limitación en el validador.

Los errores relacionados con un CodeCatalyst recurso, como secretos, entornos o flotas, en la configuración del flujo de trabajo no se registrarán durante la confirmación. Si se utiliza alguna referencia que no sea válida, el error solo se identificará cuando se ejecute un flujo de trabajo. Del mismo modo, si hay algún error en la configuración de la acción, como la falta de un campo obligatorio o errores tipográficos en los atributos de la acción, solo se identificará cuando se ejecute el flujo de trabajo. Para obtener más información, consulte [Crear, editar y eliminar un flujo de trabajo](#).

La detección automática no detecta ningún informe sobre mi acción

Problema: configuré la detección automática para una acción que ejecuta pruebas, pero no detecta ningún informe. CodeCatalyst

Posibles soluciones: esto puede deberse a varios problemas. Pruebe una o más de las siguientes soluciones:

- Asegúrese de que la herramienta utilizada para ejecutar las pruebas produzca resultados en uno de los formatos más CodeCatalyst adecuados. Por ejemplo, si desea `pytest` CodeCatalyst permitir la detección de informes de cobertura de código y pruebas, incluya los siguientes argumentos:

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Para obtener más información, consulte [Tipos de informes de prueba](#).

- Asegúrese de que la extensión del archivo de salida sea coherente con el formato elegido. Por ejemplo, al configurar `pytest` para producir resultados en `JUnitXML` formato, compruebe que la extensión del archivo sea `.xml`. Para obtener más información, consulte [Tipos de informes de prueba](#).

- Asegúrese de que `IncludePaths` está configurado para incluir todo el sistema de archivos (`**/*`), a menos que excluya determinadas carpetas a propósito. Del mismo modo, asegúrese de `ExcludePaths` no excluir los directorios en los que espera que estén ubicados sus informes.
- Si configuró manualmente un informe para usar un archivo de salida específico, se excluirá de la detección automática. Para obtener más información, consulte [Ejemplo: configurar informes](#).
- Es posible que la detección automática no encuentre los informes porque la acción falló antes de que se generara ningún resultado. Por ejemplo, es posible que la compilación haya fallado antes de ejecutar las pruebas unitarias.

Mi acción falla en los informes detectados automáticamente después de configurar los criterios de éxito

Problema: cuando habilito la detección automática y configuro los criterios de éxito, algunos de los informes no cumplen los criterios de éxito y, como resultado, la acción falla.

Posibles soluciones: para solucionar este problema, pruebe una o varias de las siguientes soluciones:

- `ExcludePaths` Modifique `IncludePaths` o excluya los informes que no le interesen.
- Actualice los criterios de éxito para permitir que se aprueben todos los informes. Por ejemplo, si se detectaron dos informes, uno con una cobertura de líneas del 50% y otro del 70%, ajuste la cobertura mínima de líneas al 50%. Para más información, consulte [Criterios correctos](#)
- Convierta el informe defectuoso en un informe configurado manualmente. Esto le permite configurar diferentes criterios de éxito para ese informe específico. Para obtener más información, consulte [Configurar los criterios de éxito de los informes](#).

La detección automática genera informes que no deseo

Problema: cuando habilito la detección automática, se generan informes que no deseo. Por ejemplo, CodeCatalyst genera informes de cobertura de código para los archivos incluidos en las dependencias de mi aplicación almacenados en `node_modules`

Posibles soluciones: puede ajustar la `ExcludePaths` configuración para excluir los archivos no deseados. Por ejemplo, para excluir `node_modules`, añadanode_modules/**/**. Para obtener más información, consulte [Incluir/excluir rutas](#).

La detección automática genera muchos informes pequeños para un único marco de pruebas

Problema: Cuando utilizo ciertos marcos de informes de cobertura de código y pruebas, observé que la detección automática genera una gran cantidad de informes. Por ejemplo, cuando se utiliza el [complemento Maven Surefire](#), la detección automática produce un informe diferente para cada clase de prueba.

Posibles correcciones: es posible que su marco pueda agregar los resultados en un solo archivo. Por ejemplo, si está utilizando el complemento Maven Surefire, puede usarlo `npx junit-merge` para agregar los archivos manualmente. La expresión completa puede tener este aspecto:

```
mvn test; cd test-package-path/surefire-reports && npx junit-merge -d ./ && rm
*Test.xml
```

Los flujos de trabajo enumerados en CI/CD no coinciden con los del repositorio de origen

Problema: [los flujos de trabajo que aparecen en la página Flujos de trabajo de CI/CD no coinciden con los de la ~/.codecatalyst/workflows/ carpeta del repositorio de origen.](#) Es posible que vea las siguientes discrepancias:

- En la página Flujos de trabajo aparece un flujo de trabajo, pero el archivo de definición de flujo de trabajo correspondiente no existe en el repositorio de origen.
- Existe un archivo de definición de flujo de trabajo en el repositorio de origen, pero el flujo de trabajo correspondiente no aparece en la página Flujos de trabajo.
- Existe un flujo de trabajo tanto en el repositorio de origen como en la página de flujos de trabajo, pero ambos son diferentes.

Este problema puede producirse si la página de flujos de trabajo no ha tenido tiempo de actualizarse o si se ha superado una cuota de flujo de trabajo.

Soluciones posibles:

- Espera. Por lo general, tienes que esperar dos o tres segundos después de confirmarlo en la fuente antes de ver el cambio en la página de flujos de trabajo.
- Si has superado una cuota de flujo de trabajo, realiza una de las siguientes acciones:

Note

Para determinar si se ha superado una cuota de flujo de trabajo [Cuotas para flujos de trabajo en CodeCatalyst](#), revisa y coteja las cuotas documentadas con los flujos de trabajo de tu repositorio de origen o de la página Flujos de trabajo. No hay ningún mensaje de error que indique que se ha superado una cuota, por lo que tendrás que investigar por tu cuenta.

- Si has superado el número máximo de flujos de trabajo por cuota de espacio, [elimina algunos flujos de trabajo](#) y, a continuación, realiza una comprobación comparándolos con el archivo de definición del flujo de trabajo. Un ejemplo de confirmación de prueba podría ser añadir un espacio al archivo.
- Si has superado la cuota máxima de tamaño de archivo de definición de flujo de trabajo, cambia el archivo de definición de flujo de trabajo para reducir su longitud.
- Si ha superado la cantidad máxima de archivos de flujo de trabajo procesados en una sola cuota de eventos de origen, realice varias confirmaciones de prueba. Modifique menos flujos de trabajo que el número máximo de flujos de trabajo en cada confirmación.
- Aumenta las cuotas de flujo de trabajo activando la facturación por niveles de pago. Para obtener más información, consulta [Gestión de la facturación](#) en la Guía del CodeCatalyst administrador de Amazon.

No puedo crear ni actualizar flujos de trabajo

Problema: quiero crear o actualizar un flujo de trabajo, pero aparece un error al intentar realizar el cambio.

Posibles soluciones: en función de tu función en el proyecto o espacio, es posible que no tengas permisos para insertar código en los repositorios de código fuente del proyecto. Los archivos YAML de los flujos de trabajo se almacenan en repositorios. Para obtener más información, consulte [Archivos de definición del flujo de trabajo](#). El rol de administrador de Space, el rol de administrador de proyectos y el rol de colaborador tienen permiso para enviar y enviar código a los repositorios de un proyecto.

Si tienes la función de colaborador, pero no puedes crear o confirmar cambios en el flujo de trabajo de YAML en una rama específica, es posible que haya una regla de ramificación configurada para

esa rama que impida a los usuarios con esa función enviar código a esa rama concreta. Intenta crear un flujo de trabajo en una rama diferente o confirmar los cambios en una rama diferente. Para obtener más información, consulte [Gestione las acciones permitidas para una sucursal con las reglas de la sucursal](#).

Solución de problemas con la búsqueda en CodeCatalyst

Consulta las siguientes secciones para solucionar problemas relacionados con las búsquedas en CodeCatalyst. Para obtener más información acerca de los flujos de trabajo, consulte [Buscar en CodeCatalyst](#).

Temas

- [No encuentro un usuario en mi proyecto](#)
- [No veo lo que busco en mi proyecto o espacio](#)
- [El número de resultados de búsqueda cambia constantemente cuando navego por las páginas](#)
- [Mi consulta de búsqueda no se está completando](#)

No encuentro un usuario en mi proyecto

Problema: cuando intento ver los detalles de un usuario, no veo su información en el proyecto.

Posibles soluciones: actualmente, la búsqueda no permite buscar usuarios dentro de un proyecto. Para buscar usuarios con acceso a tu espacio, cambia a Este espacio en QuickSearch o elimina los filtros de proyecto que hayas especificado mediante el lenguaje de consultas avanzadas.

No veo lo que busco en mi proyecto o espacio

Problema: los resultados no aparecen cuando intento buscar información concreta.

Posibles soluciones: es probable que las actualizaciones de contenido tarden unos segundos en aparecer en los resultados de búsqueda. Las actualizaciones de gran tamaño pueden tardar varios minutos.

En el caso de los recursos que no se han actualizado recientemente, puede que tengas que afinar la búsqueda. Puede refinar agregando más palabras clave o utilizando el lenguaje de consulta avanzada. Para obtener más información sobre cómo refinar las consultas, consulte [Refinar tu consulta de búsqueda](#).

El número de resultados de búsqueda cambia constantemente cuando navego por las páginas

Problema: el número de resultados de la búsqueda parece cambiar cuando paso a la página siguiente, por lo que no está claro cuántos resultados hay en total.

Posibles soluciones: al navegar por las páginas de resultados de búsqueda, es posible que veas un cambio en el número de resultados de búsqueda que coinciden con tu consulta. Es posible que el número de resultados se actualice para reflejar un número más preciso de coincidencias descubiertas al navegar por las páginas.

A medida que navega por los resultados, es posible que aparezca el siguiente mensaje: No hay resultados para la «prueba». Recibirás el mensaje si no tienes acceso al resto de los resultados.

Mi consulta de búsqueda no se está completando

Problema: los resultados de mi consulta de búsqueda no aparecen y parece que está tardando demasiado.

Soluciones posibles: es posible que la búsqueda no se complete si se están realizando muchas búsquedas al mismo tiempo en el espacio, ya sea mediante programación o debido a la alta actividad del equipo. Si realizas búsquedas programáticas, ponlas en pausa o redúcelas. De lo contrario, vuelve a intentarlo en unos segundos.

Solución de problemas con las cuentas asociadas a tu espacio

En CodeCatalyst, puedes añadir un espacio Cuenta de AWS a tu espacio para conceder permisos a los recursos y con fines de facturación. La siguiente información puede ayudarte a solucionar problemas comunes con las cuentas asociadas en CodeCatalyst.

Temas

- [Mi solicitud de Cuenta de AWS conexión recibe un error de token no válido](#)
- [CodeCatalyst El flujo de trabajo de mi proyecto de Amazon falla debido a un error en la cuenta, el entorno o el rol de IAM configurados](#)
- [Necesito una cuenta, un rol y un entorno asociados para crear un proyecto](#)
- [No puedo acceder a la página de Amazon CodeCatalyst Spaces en el AWS Management Console](#)
- [Quiero una cuenta diferente a la de facturación](#)

Mi solicitud de Cuenta de AWS conexión recibe un error de token no válido

Problema: al crear una solicitud de conexión con un token de conexión, la página no lo acepta y muestra un error que indica que el token no es válido.

Soluciones posibles: asegúrate de proporcionar el ID de cuenta que quieres añadir a tu espacio. Debe tener permisos administrativos Cuenta de AWS o poder trabajar con su administrador para añadir la cuenta.

Cuando decida verificar la cuenta, se abrirá una nueva ventana del navegador enAWS Management Console. Es necesario iniciar sesión en la misma cuenta en la consola. Vuelva a intentarlo tras comprobar lo siguiente:

- Has iniciado sesión en el mismo AWS Management Console con el Cuenta de AWS que deseas añadir a tu espacio.
- Ha iniciado sesión en el AWS Management Console con la opción EE.UU. Oeste (Oregón) Región de AWS (us-west-2) seleccionada.
- Si has llegado desde la página de facturación y quieres añadirla Cuenta de AWS como cuenta de facturación específica para tu espacio, asegúrate de que la cuenta no sea ya una cuenta de facturación de otro espacio.

CodeCatalyst El flujo de trabajo de mi proyecto de Amazon falla debido a un error en la cuenta, el entorno o el rol de IAM configurados

Problema: cuando el flujo de trabajo se ejecuta y no encuentra una cuenta configurada o funciones de IAM asociadas a su espacio, debe rellenar los campos de función, conexión y entorno manualmente en el flujo de trabajo YAML. Consulta la acción del flujo de trabajo fallida y observa si los mensajes de error son los siguientes:

- El rol no está disponible para su uso con la conexión asociada al entorno.
- La acción no se realizó correctamente. Estado: ERROR; el valor proporcionado para la conexión o el entorno de la cuenta no es válido. Compruebe que la conexión esté asociada a su espacio y que el entorno esté asociado a su proyecto.
- La acción no tuvo éxito. Estado: ERROR; el valor proporcionado para el rol de IAM no es válido. Comprueba que el nombre existe, que la función de IAM se haya añadido a la conexión de tu cuenta y que la conexión ya esté asociada a tu espacio de Amazon CodeCatalyst

Posibles correcciones: [asegúrese de que los campos YAML del flujo de trabajo tengan valores precisos para el entorno, las conexiones y el rol](#). Las acciones del CodeCatalyst flujo de trabajo que requieren un entorno son las acciones de creación o despliegue que ejecutan AWS recursos o que generan pilas AWS de recursos.

Elija el bloque de acciones del flujo de trabajo fallido y, a continuación, elija Visual. Elija la pestaña Configuración. Si los campos Entorno, Nombre de conexión y Nombre de rol no están rellenos, tendrá que actualizar manualmente el flujo de trabajo. Usa los siguientes pasos para editar tu flujo de trabajo en YAML:

- Expanda el `/.codecatalyst` directorio y, a continuación, amplíe el `/workflows` directorio. Abra el archivo YAML del flujo de trabajo. Asegúrese de que las funciones de IAM y la información de la cuenta estén especificadas en el YAML que ha configurado para su flujo de trabajo. Ejemplo:

```
Actions:
  cdk_bootstrap:
    Identifier: action-@v1
    Inputs:
      Sources:
        - WorkflowSource
    Environment:
      Name: Staging
      Connections:
        - Name: account-connection
          Role: build-role
```

Las propiedades de entorno, conexión y rol son necesarias para ejecutar acciones de creación e implementación del CodeCatalyst flujo de trabajo con AWS recursos. Para ver un ejemplo, consulta los parámetros YAML de referencia de la acción de CodeCatalyst compilación para [el entorno](#), [las conexiones](#) y el [rol](#).

- Asegúrese de que su espacio tenga una cuenta agregada y asegúrese de que la cuenta tenga el rol o roles de IAM adecuados agregados a la cuenta. Puedes ajustar o añadir cuentas si tienes el rol de administrador de Space. Para obtener más información, consulte [Cuentas de AWS Administrar un espacio](#).

Necesito una cuenta, un rol y un entorno asociados para crear un proyecto

Problema: en las opciones de creación de proyectos, mi proyecto no tiene una cuenta adicional disponible en mi espacio o necesito añadir otra cuenta a mi espacio para que la use mi proyecto.

Posibles soluciones: para tu espacio, puedes añadir personas autorizadas Cuentas de AWS para añadirlas a tu proyecto si tienes el rol de administrador del espacio. También debe tener un Cuenta de AWS lugar en el que tenga permisos administrativos o pueda trabajar con su AWS administrador.

Para asegurarse de que haya una cuenta y un rol disponibles en la pantalla de creación del proyecto, primero debe agregar la cuenta y los roles. Para obtener más información, consulte [Cuentas de AWS Administrar un espacio](#).

Tiene la opción de elegir crear un rol de servicio con una política de roles denominada política de CodeCatalystWorkflowDevelopmentRole-*spaceName*roles. El rol tendrá un nombre CodeCatalystWorkflowDevelopmentRole-*spaceName* con un identificador único adjunto. Para obtener más información sobre el rol y la política de roles, consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName*de servicio](#). Para conocer los pasos para crear el rol, consulte [Crear el CodeCatalystWorkflowDevelopmentRole-*spaceName*rol para su cuenta y su espacio](#). El rol se añade a su cuenta y está disponible en las páginas de creación de proyectos de CodeCatalyst.

No puedo acceder a la página de Amazon CodeCatalyst Spaces en el AWS Management Console

Problema: Cuando intento acceder a la CodeCatalyst página de Amazon en AWS Management Console para añadir una cuenta a mi CodeCatalyst espacio o añadir funciones a una cuenta enAWS, recibo un error de permisos.

Soluciones posibles:

Para tu espacio, puedes añadir permisos Cuentas de AWS para añadirlos a tu proyecto si tienes el rol de administrador del espacio. También debe tener un Cuenta de AWS lugar en el que tenga permisos administrativos o pueda trabajar con su AWS administrador. Primero debes asegurarte de haber iniciado sesión en ella AWS Management Console con la misma cuenta que deseas administrar. Una vez que haya iniciado sesión enAWS Management Console, podrá abrir la consola e intentarlo de nuevo.

Abre la CodeCatalyst página de Amazon AWS Management Console en <https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#/>.

Quiero una cuenta diferente a la de facturación

Problema: Cuando configuré mi CodeCatalyst nombre de usuario, realicé varios pasos para configurar mi espacio y asociar a un usuario autorizado Cuenta de AWS. Ahora, quiero autorizar la facturación a otra cuenta.

Posibles soluciones: para tu espacio, puedes autorizar las cuentas de facturación si tienes el rol de administrador del espacio. También debes tener un Cuenta de AWS lugar en el que tengas permisos administrativos o puedas trabajar con AWS él.

Para obtener más información, consulta [Gestión de la facturación](#) en la Guía del CodeCatalyst administrador de Amazon.

Solución de problemas con entornos de desarrollo

Consulte las siguientes secciones para solucionar problemas relacionados con los entornos de desarrollo. Para obtener más información sobre los entornos de desarrollo, consulte [Entornos de desarrollo en CodeCatalyst](#).

Temas

- [La creación de mi entorno de desarrollo no se ha realizado correctamente debido a un problema con las cuotas](#)
- [No puedo enviar cambios de mi entorno de desarrollo a una rama específica de un repositorio](#)
- [Mi entorno de desarrollo no se reanudó](#)
- [Mi entorno de desarrollo se ha desconectado](#)
- [Falló mi entorno de desarrollo conectado a VPC](#)
- [No encuentro el directorio en el que se encuentra mi proyecto](#)
- [No puedo conectarme a mi entorno de desarrollo a través de SSH](#)
- [No puedo conectarme a mi entorno de desarrollo a través de SSH porque falta mi configuración de SSH local](#)
- [No puedo conectarme a mi entorno de desarrollo a través de SSH porque tengo problemas con mi perfil AWS Configcodecatalyst](#)
- [Solución de problemas con los IDE](#)
- [Solución de problemas con devfiles](#)

La creación de mi entorno de desarrollo no se ha realizado correctamente debido a un problema con las cuotas

Problema: quiero crear un entorno de desarrollo en CodeCatalyst, pero aparece un error. En la consola, veo un mensaje en la página de entornos de desarrollo que indica que he alcanzado el límite de almacenamiento del espacio.

Soluciones posibles: en función del rol que desempeñes en el proyecto o espacio, puedes eliminar uno o más de tus propios entornos de desarrollo o, si tienes el rol de administrador del espacio, puedes eliminar los entornos de desarrollo no utilizados creados por otros usuarios. También puedes decidir cambiar el nivel de facturación por uno que incluya más espacio de almacenamiento.

- Para ver los límites de almacenamiento, consulta la pestaña Facturación del CodeCatalyst espacio de Amazon para comprobar si las cuotas de uso han alcanzado el máximo permitido. Si las cuotas han alcanzado el máximo, ponte en contacto con alguien que desempeñe el rol de administrador de espacios para que elimine los entornos de desarrollo innecesarios o considere la posibilidad de cambiar el nivel de facturación.
- Para eliminar cualquier entorno de desarrollo que haya creado y que ya no necesite, consulte [Eliminación de un entorno de desarrollo](#).

Si el problema persiste y aparece un error en el IDE, compruebe que tiene un CodeCatalyst rol que le permita crear un entorno de desarrollo. El rol de administrador del espacio, el rol de administrador de proyectos y el rol de colaborador tienen permiso para crear entornos de desarrollo. Para obtener más información, consulte [Trabajar con roles en Amazon CodeCatalyst](#).

No puedo enviar cambios de mi entorno de desarrollo a una rama específica de un repositorio

Problema: quiero confirmar y enviar los cambios de código de mi entorno de desarrollo a una rama de un repositorio de código fuente, pero aparece un error.

Soluciones posibles: en función de tu función en el proyecto o espacio, es posible que no tengas permisos para insertar código en los repositorios de código fuente del proyecto. El rol de administrador de Space, el rol de administrador del proyecto y el rol de colaborador tienen permiso para insertar código en los repositorios del proyecto.

Si tienes el rol de colaborador pero no puedes enviar código a una rama específica, es posible que haya una regla de ramificación configurada para esa rama específica que impida a los usuarios con

ese rol enviar código a esa rama en particular. Intenta enviar los cambios a una rama diferente o crea una rama y, a continuación, envía el código a esa rama. Para obtener más información, consulte [Gestione las acciones permitidas para una sucursal con las reglas de la sucursal](#).

Mi entorno de desarrollo no se reanudó

Problema: mi entorno de desarrollo no se reanudó después de detenerlo.

Posibles soluciones: para solucionar el problema, consulta la pestaña Facturación del CodeCatalyst espacio de Amazon para comprobar si las cuotas de uso han alcanzado los límites máximos. Si las cuotas han alcanzado el límite máximo, ponte en contacto con el administrador de Space para aumentar el nivel de facturación.

Mi entorno de desarrollo se ha desconectado

Problema: mi entorno de desarrollo se desconectó mientras lo estaba usando.

Posibles soluciones: para solucionar el problema, comprueba tu conexión a Internet. Si no está conectado a Internet, conéctese y vuelva a trabajar en su entorno de desarrollo.

Falló mi entorno de desarrollo conectado a VPC

Problema: he asociado una conexión de VPC a mi entorno de desarrollo y se producen errores.

Soluciones posibles: Docker utiliza un dispositivo de capa de enlace denominado red puente que permite la comunicación de los contenedores que están conectados a la misma red puente. El puente predeterminado suele utilizar la subred 172.17.0.0/16 para las redes de contenedores. Si la subred de VPC para la instancia del entorno utiliza el mismo rango de direcciones que Docker, es posible que se produzca un conflicto de dirección IP. Para resolver un conflicto de direcciones IP provocado por Amazon VPC y Docker usar el mismo bloque de direcciones CIDR de IPv4, configure un bloque de CIDR diferente de 172.17.0.0/16

Note

No puedes cambiar el rango de direcciones IP de una VPC o subred existente.

No encuentro el directorio en el que se encuentra mi proyecto

Problema: no encuentro en qué directorio se encuentra mi proyecto.

Posibles soluciones: para localizar su proyecto, cambie el directorio a `projects`. Este es el directorio donde puedes encontrar tus proyectos.

No puedo conectarme a mi entorno de desarrollo a través de SSH

Para solucionar los problemas de conexión a tu entorno de desarrollo a través de SSH, puedes ejecutar el `ssh` comando con la `-vvv` opción de mostrar más información sobre cómo resolver el problema:

```
ssh -vvv codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

No puedo conectarme a mi entorno de desarrollo a través de SSH porque falta mi configuración de SSH local

Si falta tu configuración de SSH local (`~/.ssh/config`) o el contenido de la `Host codecatalyst-dev-env*` sección no está actualizado, no podrás conectarte a tu entorno de desarrollo a través de SSH. Para solucionar este problema, elimina la `Host codecatalyst-dev-env*` sección y vuelve a ejecutar el primer comando desde el modal SSH Access. Para obtener más información, consulte [Conexión a un entorno de desarrollo mediante SSH](#).

No puedo conectarme a mi entorno de desarrollo a través de SSH porque tengo problemas con mi perfil AWS Config **codecatalyst**

Asegúrese de que su AWS Config (`~/.aws/config`) del `codecatalyst` perfil coincida con el que se describe en [Configuración para usar el AWS CLI con CodeCatalyst](#). Si no es así, elimine el perfil `codecatalyst` y vuelva a ejecutar el primer comando desde el modal SSH Access. Para obtener más información, consulte [Conexión a un entorno de desarrollo mediante SSH](#).

Solución de problemas con los IDE

Consulta las siguientes secciones para solucionar problemas relacionados con los IDE en CodeCatalyst. Para obtener más información sobre los IDE, consulte [Uso de entornos de desarrollo con un IDE](#).

Temas

- [Tengo versiones de imágenes en tiempo de ejecución que no coinciden en AWS Cloud9](#)
- [No puedo acceder a mis archivos `/projects/projects` en AWS Cloud9](#)
- [No puedo iniciar mi entorno de desarrollo AWS Cloud9 con un archivo de desarrollo personalizado](#)

- [Tengo problemas en AWS Cloud9](#)
- [En JetBrains, no puedo conectarme a mis entornos de desarrollo a través de CodeCatalyst](#)
- [No puedo instalarlo AWS Toolkit para mi IDE](#)
- [En mi IDE, no puedo lanzar mis entornos de desarrollo](#)

Tengo versiones de imágenes en tiempo de ejecución que no coinciden en AWS Cloud9

AWS Cloud9 está utilizando diferentes versiones del recurso de interfaz y de la imagen de tiempo de ejecución de fondo. El uso de versiones diferentes puede provocar que la extensión Git no funcione correctamente. AWS Toolkit Para solucionar el problema, dirígete al panel de control del entorno de desarrollo, detiene el entorno de desarrollo y vuelve a iniciarlo. Para solucionar el problema con las API, usa la `UpdateDevEnvironment` API para actualizar el tiempo de ejecución. Para obtener más información, consulta [UpdateDevEnvironment](#) la referencia de la CodeCatalyst API de Amazon.

No puedo acceder a mis archivos `/projects/projects` en AWS Cloud9

El AWS Cloud9 editor no puede acceder a los archivos del directorio `/projects/projects`. Para solucionar el problema, utilice el AWS Cloud9 terminal para acceder a sus archivos o muévalos a un directorio diferente.

No puedo iniciar mi entorno de desarrollo AWS Cloud9 con un archivo de desarrollo personalizado

Es posible que tu imagen de devfile no sea compatible con. AWS Cloud9 Para solucionar el problema, revisa el devfile de tu repositorio y el entorno de desarrollo correspondiente y crea uno nuevo para continuar.

Tengo problemas en AWS Cloud9

Para otros problemas, consulta la sección de solución de problemas de la [Guía del AWS Cloud9 usuario](#).

En JetBrains, no puedo conectarme a mis entornos de desarrollo a través de CodeCatalyst

Para solucionar el problema, comprueba que solo tienes JetBrains instalada la última versión de. Si tiene varias versiones, desinstale las versiones anteriores y vuelva a registrar el controlador de

protocolos cerrando el IDE y el navegador. A continuación, abra JetBrains y registre de nuevo el controlador de protocolos.

No puedo instalarlo AWS Toolkit para mi IDE

Para solucionar este problema en VS Code, instálelo manualmente AWS Toolkit for Visual Studio Code desde [GitHub](#).

Para solucionar este problema JetBrains, instálelo manualmente AWS Toolkit for JetBrains desde [GitHub](#).

En mi IDE, no puedo lanzar mis entornos de desarrollo

Para solucionar este problema en VS Code, comprueba que tienes AWS Toolkit for Visual Studio Code instalada la última versión de VS Code. Si no tienes la última versión, actualiza e inicia tu entorno de desarrollo. Para obtener más información, consulta [Amazon CodeCatalyst for VS Code](#).

Para solucionar este problema JetBrains, comprueba que tienes la última versión de JetBrains e AWS Toolkit for JetBrains instalada. Si no tienes la versión más reciente, actualiza e inicia tu entorno de desarrollo. Para obtener más información, consulte [Amazon CodeCatalyst for JetBrains](#).

Solución de problemas con devfiles

Consulta las siguientes secciones para solucionar problemas relacionados con los archivos de desarrollo en. CodeCatalyst Para obtener más información sobre devfiles, consulte. [Configuración de su entorno de desarrollo](#)

Temas

- [Mi entorno de desarrollo utiliza el archivo de desarrollo universal predeterminado, aunque he implementado una imagen personalizada en un archivo de desarrollo personalizado](#)
- [Mi proyecto no se está compilando en mi entorno de desarrollo con el archivo de desarrollo universal predeterminado](#)
- [Quiero mover un archivo de desarrollo del repositorio a un entorno de desarrollo](#)
- [Tengo problemas para iniciar mi devfile](#)
- [No estoy seguro de cómo comprobar el estado de mi devfile](#)
- [Mi archivo de desarrollo no es compatible con las herramientas proporcionadas en la última imagen](#)

Mi entorno de desarrollo utiliza el archivo de desarrollo universal predeterminado, aunque he implementado una imagen personalizada en un archivo de desarrollo personalizado

Si CodeCatalyst se producen errores al iniciar un entorno de desarrollo que utiliza un archivo de desarrollo personalizado, el entorno de desarrollo utilizará de forma predeterminada el archivo de desarrollo universal predeterminado. Para solucionar el problema, puedes comprobar el error exacto en los registros de abajo. `/aws/mde/logs/devfile.log` También puede comprobar si `postStart` la ejecución se ha realizado correctamente en sus registros: `/aws/mde/logs/devfileCommand.log`.

Mi proyecto no se está compilando en mi entorno de desarrollo con el archivo de desarrollo universal predeterminado

Para solucionar el problema, compruebe que no está utilizando un archivo de desarrollo personalizado. Si no está utilizando un `devfile` personalizado, visualice el `devfile.yaml` archivo en el repositorio fuente del proyecto para localizar y corregir cualquier error.

Quiero mover un archivo de desarrollo del repositorio a un entorno de desarrollo

Puedes mover el archivo de desarrollo predeterminado `/projects/devfile.yaml` a tu repositorio de código fuente. Para actualizar la ubicación del `devfile`, usa el siguiente comando: `/aws/mde/mde start --location repository-name/devfile.yaml`

Tengo problemas para iniciar mi devfile

Si se produce algún problema al iniciar tu `devfile`, pasará al modo de recuperación para que puedas seguir conectándote a tu entorno y corregir tu `devfile`. Mientras esté en modo de recuperación, al ejecutar `/aws/mde/mde status` no se incluirá la ubicación del archivo de desarrollo.

```
{
  "status": "STABLE"
}
```

Puedes comprobar el error en los registros de abajo `/aws/mde/logs`, corregir el `devfile` e intentar volver a `/aws/mde/mde start` ejecutarlo.

No estoy seguro de cómo comprobar el estado de mi devfile

Puede comprobar el estado de su archivo de desarrollo ejecutándolo. `/aws/mde/mde status` Tras ejecutar este comando, es posible que veas una de las siguientes opciones:

- `{"status": "STABLE", "location": "devfile.yaml" }`

Esto indica que el archivo devfile es correcto.

- `{"status": "STABLE" }`

Esto indica que su devfile no se ha podido iniciar y ha entrado en modo de recuperación.

Puedes comprobar el error exacto en los registros de abajo `/aws/mde/logs/devfile.log`.

También puede comprobar si `postStart` la ejecución se realizó correctamente en sus registros: `/aws/mde/logs/devfileCommand.log`.

Para obtener más información, consulte [Imágenes universales devfile](#).

Mi archivo de desarrollo no es compatible con las herramientas proporcionadas en la última imagen

En su entorno de desarrollo, `devfile` o `devfile postStart` puede fallar si las `latest` herramientas no tienen las herramientas necesarias para un proyecto específico. Para solucionar el problema, haga lo siguiente:

1. Navega hasta tu archivo devfile.
2. En tu devfile, actualiza a una versión de imagen granular en lugar de `latest`. Puede tener un aspecto similar al siguiente:

```
components:
  - container:
      image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Cree un nuevo entorno de desarrollo con el devfile actualizado.

Solución de problemas con problemas

La siguiente información puede ayudarte a solucionar problemas comunes relacionados con problemas de CodeCatalyst

Temas

- [No puedo elegir un responsable para mi problema](#)

No puedo elegir un responsable para mi problema

Problema: Al crear una emisión, la lista de cesionarios está vacía.

Posibles soluciones: la lista de personas asignadas está directamente vinculada a los CodeCatalyst usuarios que figuran como miembros del proyecto. Para comprobar que el acceso al perfil de usuario funciona correctamente, elija el icono del perfil y, a continuación, elija el perfil de usuario. Si la información del perfil de usuario no aparece, consulte el informe de estado para ver si hay algún incidente. Si se completa, presente una solicitud de servicio.

Solución de problemas entre Amazon CodeCatalyst y los AWS SDK o el AWS CLI

La siguiente información puede ayudarte a solucionar problemas habituales al trabajar con AWS CLI o con CodeCatalyst los SDK. AWS

Temas

- [Recibo un mensaje de error cuando introduzco en la línea de comandos o aws codecatalyst en el terminal el mensaje de error que indica que la elección no es válida](#)
- [Recibo un error de credenciales al ejecutar aws codecatalyst comandos](#)

Recibo un mensaje de error cuando introduzco en la línea de comandos o aws codecatalyst en el terminal el mensaje de error que indica que la elección no es válida

Problema: Cuando intento usar el comando AWS CLI with CodeCatalyst, uno o más de los aws codecatalyst comandos no se reconocen como válidos.

Solución: la causa más común de este problema es que está utilizando una versión de la AWS CLI que no contiene las actualizaciones más recientes de los servicios y comandos más recientes. Actualice la instalación del AWS CLI e inténtelo de nuevo. Para obtener más información, consulte [Configuración para usar el AWS CLI con CodeCatalyst](#).

Recibo un error de credenciales al ejecutar aws codecatalyst comandos

Problema: Cuando intento usar el comando AWS CLI with CodeCatalyst, recibo un mensaje que indica `You can configure credentials by running "aws configure". Unable to locate authorization token.`

Solución: debe configurar un AWS CLI perfil para trabajar con CodeCatalyst los comandos. Para obtener más información consulte [Configuración para usar el AWS CLI con CodeCatalyst](#).

CodeCatalyst informe de salud

El informe de CodeCatalyst salud de Amazon es un panel público que proporciona a los usuarios una lista agregada de up-to-the-minute notificaciones sobre el rendimiento de los recursos y la disponibilidad de los servicios CodeCatalyst que tienen un impacto generalizado. Puede ver qué recursos tienen problemas y cuáles pueden afectar a las aplicaciones CodeCatalyst. Esto le permite realizar un seguimiento de las interrupciones y otros tiempos de inactividad de los recursos en todo el sistema. Cuando se produce un incidente, aparece un indicador azul en el icono del informe de estado. Además, envía CodeCatalyst automáticamente una alerta y una notificación por correo electrónico a todos los usuarios que tengan la función de administrador de Space en el proyecto, proporcionando detalles y un historial del incidente prácticamente en tiempo real.

El panel de control proporciona una lista de todos los eventos activos y un registro de hasta 100 incidentes anteriores ocurridos en los últimos 30 días. Puede organizar la lista de incidentes en función de la fecha en que se actualizó el incidente. También puede actualizar la lista de incidentes para obtener actualizaciones al minuto.

Este es un posible flujo de trabajo para usar el informe CodeCatalyst de estado:

Mateo Jackson es un desarrollador de Budding Space con permisos de administrador de Space. Al intentar crear una solicitud de cambios, sigue recibiendo un mensaje de error. Comprueba su correo electrónico y descubre que ha recibido un correo electrónico de incidente del sistema generado automáticamente CodeCatalyst al proporcionar un historial detallado sobre el problema del sistema que afecta a su espacio. Selecciona Ver actualización y accederá al informe de CodeCatalyst estado, donde podrá ver todos los incidentes notificados por el sistema. Selecciona el incidente de la lista para obtener más información. Se abre una pantalla dividida que muestra la fecha y hora de la última actualización, el historial, las capacidades afectadas, la hora de inicio y el estado actual del incidente. También puede ver que el problema continúa, pero el equipo de servicio ha empezado a trabajar en ello. Cada vez que se actualiza el historial o el estado del incidente, recibe un correo electrónico. En caso de que no tenga acceso a su correo electrónico, puede elegir el icono de la campana en el panel superior para acceder al informe de CodeCatalyst salud.

CodeCatalyst conceptos de informes de salud

Aprender los siguientes conceptos le ayudará a entender el informe de CodeCatalyst estado y cómo le permitirán realizar un seguimiento del estado de sus aplicaciones, servicios y recursos.

Incidente

El incidente es el suceso del sistema que afecta a las aplicaciones y los recursos internos CodeCatalyst. Puede elegir el incidente para ver un historial detallado del evento, incluida la hora en que comenzó y si el equipo de servicio está trabajando para resolverlo.

Estado

El estado es el estado del incidente en tiempo real. Se mostrará como En curso o Resuelto.

Capacidades afectadas

Las capacidades afectadas son los recursos o las aplicaciones afectados por el incidente. Un solo incidente puede afectar a varias áreas del sistema, incluidas las solicitudes de cambios, los problemas, los flujos de trabajo, las pruebas, el despliegue y el origen.

Actualizado el

Actualizado el proporciona una marca de tiempo de la última actualización del incidente.

AWS Support para Amazon CodeCatalyst

Al crear un espacio, debe conectar un espacio Cuenta de AWS y designarlo como la cuenta de facturación de su espacio. La cuenta Cuenta de AWS que designes como tu cuenta de facturación también es donde accedes a tu AWS Support plan para Amazon CodeCatalyst. Si necesitas asistencia, puedes crear casos de asistencia a partir de esta opción designada Cuenta de AWS.

CodeCatalyst los usuarios de un espacio utilizan la CodeCatalyst página de Amazon AWS Support CodeCatalyst para gestionar los casos de soporte. Puede actualizar a un AWS Support plan como Business Support o Enterprise Support para crear y gestionar casos de soporte CodeCatalyst técnico en CodeCatalyst. El soporte está disponible por teléfono, web o chat para casos de soporte.

Amazon solo admite casos específicos del CodeCatalyst servicio y AWS Support los recursos CodeCatalyst. CodeCatalyst los recursos incluyen los recursos desplegados dentro CodeCatalyst y por los usuarios en CodeCatalyst, pero no incluyen los recursos implementados para servicios AWS ajenos o de terceros. Si necesita soporte para cualquier otro AWS servicio, debe abrirlo a través del AWS Management Console.

Para cambiar su plan de soporte, consulte [Cambiar los planes de soporte](#).

Note

Los planes de Developer Support no están diseñados para entornos de producción. Si una cuenta de facturación de espacios tiene un plan Developer Support, este plan no se extiende en cascada a todos los administradores del espacio ni a los miembros del espacio que lo integran AWS Support. CodeCatalyst


Facturación AWS Support para Amazon CodeCatalyst

Al crear un espacio en CodeCatalyst, los usuarios del espacio pueden crear y gestionar casos de soporte desde AWS Support Amazon CodeCatalyst. Puedes crear dos tipos de casos de clientes:

- Los casos de soporte de cuentas y facturación están disponibles para todos CodeCatalyst los usuarios del espacio. Puedes obtener ayuda con las preguntas sobre facturación y cuenta en función de tus permisos en CodeCatalyst.


- Los casos de soporte técnico lo ponen en contacto con un ingeniero de soporte técnico para que lo ayude con los problemas técnicos relacionados con el servicio y con las extensiones de aplicaciones de terceros. Si tiene el plan de soporte Basic, no puede crear un caso de soporte técnico.

La cuenta Cuenta de AWS designada como cuenta de facturación del espacio debe tener un plan Business Support o Enterprise Support para el espacio a utilizar AWS Support en casos técnicos. CodeCatalyst

 Note

Si tu espacio se utiliza AWS Support para Amazon CodeCatalyst desde una cuenta que no tiene un plan Business Support o Enterprise Support, puedes seguir utilizándolo AWS Support para Amazon CodeCatalyst para casos de cuenta y facturación.

Para obtener asistencia técnica, debes abrir todos los casos a través de la CodeCatalyst consola. No puede crear casos de soporte técnico CodeCatalyst desde [AWS Support](#) AWS Management Console.

 Note

Las solicitudes de aumento del límite de AWS Support servicio no están disponibles en Amazon CodeCatalyst. Estas solicitudes solo las puede enviar el usuario raíz de la cuenta de facturación de espacios de AWS Support Center Console.

AWS Support porque Amazon CodeCatalyst tiene los mismos acuerdos de soporte que AWS Support, con las siguientes consideraciones:

- Las listas de gravedad, los tiempos de respuesta y los acuerdos de nivel de servicio AWS Support aplicables a los casos de soporte CodeCatalyst, tal y como se detalla en [Cómo elegir un nivel de gravedad](#). AWS Support
- Los administradores y los miembros del espacio no pueden usar las AWS Support API, el AWS SDK o AWS Support la aplicación de Slack para crear casos. CodeCatalyst CodeCatalyst los casos de soporte solo se pueden enviar desde CodeCatalyst.

Note

CodeCatalyst no está completamente integrado con AWS Trusted Advisor la detección y respuesta a AWS incidentes. Valide cómo CodeCatalyst se integra para garantizar que sus prácticas empresariales estén alineadas con la integración actual.

Debe ser un usuario del espacio en el que desea solicitar asistencia.

Note


Si tiene más de un constructor en su espacio, le recomendamos que compre un plan Business Support o Enterprise Support. Estos planes proporcionan asistencia técnica para el espacio a un máximo de 5000 constructores.

La cuenta de AWS designada como cuenta de facturación del espacio utiliza el `AWSRoleForCodeCatalystSupport` rol y la política [AmazonCodeCatalystSupportAccess](#) gestionada. Esto permite a CodeCatalyst los usuarios de un espacio acceder a AWS Support la CodeCatalyst página de Amazon. Para obtener más información sobre este rol y esta política, consulte [AmazonCodeCatalystSupportAccess](#). Para otras consideraciones sobre la facturación, consulta [Gestión de la facturación](#) en la Guía del CodeCatalyst administrador de Amazon.

Este es un posible flujo para que un constructor cree un caso de soporte en CodeCatalyst:

Mateo Jackson es desarrollador de un proyecto en CodeCatalyst. Tras suscribirse a la empresa Cuenta de AWS que gestiona la facturación AWS Support de Amazon CodeCatalyst y pasarse a un plan Business Support, todos los constructores del sector pueden crear casos de soporte técnico. Mateo presenta un caso de soporte técnico por un flujo de trabajo fallido en su proyecto. Mateo utiliza la CodeCatalyst página de Amazon AWS Support para rellenar el formulario y crear un caso, proporcionando el identificador del flujo de trabajo y otros detalles en la solicitud. El caso se crea con un identificador de caso e incluye el identificador de cuenta de la cuenta de AWS designada como cuenta de facturación y está asociado al plan de asistencia del espacio.

Si bien todos los creadores pueden crear casos AWS Support de soporte CodeCatalyst, no se cobrará por cada caso creado. Puedes abrir casos y contactos prácticamente ilimitados en función del plan AWS Support Premium que compres en tu cuenta de facturación de espacios.

 Note


La cuenta de facturación de espacio es la Cuenta de AWS que se te cobra por CodeCatalyst los usuarios y los recursos. Si ha desplegado otros servicios Cuentas de AWS, póngase en contacto AWS Support con nosotros AWS Management Console para obtener ayuda con los recursos desplegados en otros servicios.

Puede identificar el elemento en el Cuenta de AWS que se implementó en el flujo de trabajo.

Configurar tu espacio AWS Support para Amazon CodeCatalyst

AWS Support for Amazon CodeCatalyst gestiona los casos de soporte como parte de la integración de la AWS Support API con CodeCatalyst.

El `AWSRoleForCodeCatalystSupport` rol es un rol de servicio que se utiliza para los casos de soporte en tu espacio. El rol debe agregarse a la cuenta de facturación designada para el espacio. Para obtener más información o para crear el rol, consulte [Crear el AWSRoleForCodeCatalystSupport rol para tu cuenta y tu espacio](#).

 Note

En el caso de un espacio que se creó antes del 20 de abril de 2023, debe crear el rol para CodeCatalyst que el soporte funcione en su espacio. Si vas a crear un espacio después del 20 de abril de 2023, puedes crear el rol durante la creación del espacio, en la página de detalles de facturación o haciendo clic en CodeCatalyst el enlace de soporte que aparece en él CodeCatalyst.

Para configurar el soporte para tu espacio

1. Al crear un CodeCatalyst espacio, se te indica que conectes una cuenta de facturación. La cuenta de facturación designada para el espacio se facturará por AWS. Para obtener más información sobre la creación de un espacio, consulte [Inscríbete para crear tu primer espacio y tu rol de desarrollo](#).
2. Al crear un CodeCatalyst espacio, está disponible la opción de crear el rol de `AWSRoleForCodeCatalystSupport` servicio que permite a CodeCatalyst los usuarios

acceder al soporte. El rol usa la política administrada `AmazonCodeCatalystSupportAccess`. El rol debe agregarse a la cuenta Cuenta de AWS designada como cuenta de facturación para el espacio. Para obtener más información sobre la creación de este rol, consulte [Crear el AWSRoleForCodeCatalystSupportrol para tu cuenta y tu espacio](#).

3. Para la cuenta de facturación designada para el espacio, se recomienda al administrador del espacio que compre un plan Business Support o Enterprise Support para elCuenta de AWS. Todos los miembros del espacio podrán gestionar los casos de soporte de Amazon CodeCatalyst, y AWS Support los canales de asistencia se ajustarán al AWS Support plan que hayas adquirido, en el que se completen las integraciones.
4. Para crear y gestionar los casos de soporte en CodeCatalyst. [Crear un caso CodeCatalyst de soporte en CodeCatalyst](#)

Acceder al soporte CodeCatalyst en el AWS Management Console

Si se desconecta la cuenta de facturación de un espacio habilitada para el soporte, AWS Support los casos asociados a la cuenta de facturación de espacio anterior y al plan de soporte asociado dejarán de estar visibles en AWS Support Amazon CodeCatalyst. El usuario raíz de esa cuenta de facturación puede ver y resolver casos antiguos AWS Management Console y configurar permisos de IAM AWS Support para que otros usuarios puedan ver y resolver casos antiguos. Podrás seguir beneficiándote de los beneficios de tu plan de soporte desde el de todos los demás Servicios de AWS y resolver cualquier caso de CodeCatalyst soporte que no se haya resuelto anteriormente.

AWS Management Console

Para obtener más información, consulte [Actualización, resolución y reapertura de un caso](#) en la Guía del AWS Supportusuario.

Los casos de soporte con información práctica general también se CodeCatalyst pueden abrir en elAWS Management Console, pero no se puede recibir asistencia técnica a través de este canal. CodeCatalyst Para obtener más información, consulte [Creación de casos de soporte y gestión de casos](#) en la Guía del AWS Supportusuario.

A continuación, se muestra un posible flujo para que un usuario resuelva un caso de soporte CodeCatalyst enAWS Management Console:

Si bien todos los creadores pueden crear casos de soporte AWS Support para Amazon CodeCatalyst, las solicitudes de soporte se facturan desde la cuenta designada como cuenta de facturación del espacio. Mateo Jackson es un desarrollador de un proyecto en el CodeCatalyst

que presentó un caso de soporte técnico debido a un flujo de trabajo fallido en su proyecto. Sin embargo, la cuenta de facturación del espacio en el que estaba registrado AWS Support en Amazon CodeCatalyst y que había comprado un plan Business Support se ha desconectado del espacio. La única forma en que Mateo puede ver las últimas comunicaciones y resolver los casos pendientes CodeCatalyst es gestionando el identificador del caso desde el AWS Support Centro delAWS Management Console. Para ello, Mateo recibe permisos de IAM del usuario raíz de la cuenta de facturación de espacios anterior adjunta a su caso de soporte y resuelve el caso AWS Support en la consola.

Important

Si cambias la cuenta de facturación designada para tu espacio, solo podrás acceder a tu AWS Support plan hasta fin de mes. AWS Management Console Deberás volver a realizar la compra AWS Support en la cuenta de facturación actualizada para poder seguir accediendo a los casos de asistencia que creaste anteriormente. CodeCatalyst Te recomendamos que esperes hasta que hayas resuelto todos tus casos de soporte para cambiar las cuentas de facturación de espacio y evitar que el acceso a tus casos de soporte a través de Amazon se AWS Support vea afectado CodeCatalyst.

Crear un caso CodeCatalyst de soporte en CodeCatalyst

Puedes crear un caso de soporte en la CodeCatalyst página AWS Support de Amazon.

Un AWS Builder ID solo puede ser compatible con el alias con el que está autenticado y solo con los recursos en función de sus permisos. Las opciones de cuenta y facturación están disponibles para todos los administradores y miembros del espacio. Sin embargo, los usuarios solo pueden obtener asistencia para los recursos a los que tienen acceso CodeCatalyst y no en relación con la administración de la facturación de la cuenta.

Puedes crear un caso de cuenta y facturación o un caso de asistencia técnica para tus CodeCatalyst recursos utilizando la AWS Support CodeCatalyst página de tu espacio.

Note

Amazon solo admite casos específicos del CodeCatalyst servicio y AWS Support los recursos CodeCatalyst. CodeCatalyst los recursos incluyen los recursos desplegados dentro CodeCatalyst y por los usuarios en CodeCatalyst, pero no incluyen los recursos

implementados para servicios AWS ajenos o de terceros. Si necesita soporte para cualquier otro AWS servicio, debe abrirlo a través del AWS Management Console.


Para crear un caso de soporte en CodeCatalyst

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

 Tip

Si perteneces a más de un espacio, elige uno en la barra de navegación superior.

3. En la parte superior de la página, selecciona ? icono y, a continuación, selecciona Support.
4. Elija Create case (Crear caso).
5. Elija una de las siguientes opciones:
 - Cuenta y facturación
 - Técnica


 Note

En el caso AWS Support de Amazon CodeCatalyst, si se añade un plan Business Support o Enterprise Support a la cuenta de facturación del espacio, el soporte CodeCatalyst técnico estará disponible para todos los administradores del espacio y los miembros del espacio. Para obtener información sobre la resolución de problemas, consulte [No puedo crear casos de soporte técnico para mi espacio](#).

AWS Support los planes no abarcan todos los espacios. Si eres miembro de varios espacios, el administrador del espacio tendrá que comprar un plan AWS Support Premium para cada espacio a fin de recibir asistencia técnica en todos los espacios.


6. Elija Service (Servicio) , Category (Categoría) y Severity (Gravedad). Para obtener información sobre cómo elegir una gravedad, consulte [Elegir una gravedad](#).
 - General guidance
 - System impaired
 - Production system impaired

- Production system down
 - Business-critical system down
7. Elija Next step: Additional information (Siguiente paso: información adicional).
 8. En la página Additional information (Información adicional), para Subject (Asunto), introduzca un título sobre su problema.
 9. En Description (Descripción), siga las instrucciones para describir su caso, como se indica a continuación:
 - Información de solución de problemas específica CodeCatalyst, como el identificador del flujo de trabajo, los registros o las capturas de pantalla
 - Mensajes de error que ha recibido
 - Pasos de solución de problemas que ha seguido

 Note

No compartas información confidencial en caso de correspondencia, como credenciales, tarjetas de crédito, direcciones URL firmadas o información de identificación personal.

10. (Opcional) Elija Attach files (Adjuntar archivos) para añadir cualquier archivo relevante a su caso, como registros de errores o capturas de pantalla. Puede adjuntar hasta tres archivos. Cada archivo puede ser de hasta 5 MB.
11. En Nombre del espacio, aparece el nombre de tu espacio.
12. En Nombre del constructor, se rellena automáticamente el nombre completo asociado a tu ID de AWS constructor.
13. (Opcional) Elija el proyecto en el nombre del proyecto (si corresponde).

 Note

Solo se te mostrarán los proyectos para los que tengas permisos. Si necesitas acceder a otro proyecto, pídele al administrador del proyecto que te dé acceso antes de crear un caso de soporte.

14. Selecciona Siguiente paso: ponte en contacto con nosotros.
15. En Idioma de contacto preferido, selecciona el predeterminado. En este momento, solo está disponible el inglés.

16. Elija la opción web, telefónica o de chat como método de contacto.
17. Revisa los detalles del caso y, a continuación, selecciona Enviar. Aparecerán el número de ID del caso y el resumen.

El caso de soporte se crea a nivel de espacio y lo pueden ver todos los miembros con acceso al espacio y al proyecto (si están seleccionados) definidos en su caso de soporte. No hay forma de omitir un caso de soporte para los usuarios individuales en este momento.

Resolver un caso de soporte en CodeCatalyst

Puedes resolver los casos de soporte pendientes desde la CodeCatalyst página AWS Support de Amazon.

Debes tener un rol de administrador o miembro de Space en el espacio en el que quieras resolver un caso de soporte. Si no tienes el rol de administrador de Space o si seleccionaste un proyecto cuando se creó el caso, también necesitarás ser miembro del proyecto para ver y resolver el caso.

Para resolver un caso de soporte pendiente en CodeCatalyst

1. Abra la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. En la parte superior de la página, selecciona la opción? icono y, a continuación, elige AWS SupportAmazon CodeCatalyst.
4. Elige el enlace del caso de soporte que quieres gestionar. Elija Resolve case (Resolver el caso).

Reabrir un caso de soporte en CodeCatalyst

Puedes usar la opción Reabrir un caso de soporte resuelto en la CodeCatalyst página AWS Support de Amazon.

Note

Puede reabrir el caso de soporte hasta 14 días desde que se resolvió el problema. Sin embargo, no puede reabrir un caso que haya estado inactivo durante más de 14 días. Si no puedes volver a abrir tu caso, abre uno nuevo e incluye el identificador del caso anterior como referencia.

Si reabre un caso existente que tiene información diferente a la del problema actual, es posible que el agente de soporte le pida que cree un caso nuevo.

Para volver a abrir un caso de soporte en CodeCatalyst

1. Abre la CodeCatalyst consola en <https://codecatalyst.aws/>.
2. Navega hasta tu CodeCatalyst espacio.

Tip

Si perteneces a más de un espacio, elige un espacio en la barra de navegación superior.

3. En la parte superior de la página, selecciona la opción? icono y, a continuación, seleccione AWS Support for CodeCatalyst.
4. Elija el enlace del caso de soporte que desee gestionar. Elija Reopen (Reabrir). Selecciona Aceptar en la pantalla de confirmación y, a continuación, selecciona Enviar.
5. Complete la descripción con la información más reciente sobre el mismo problema. No compartas información confidencial en caso de correspondencia, como credenciales, tarjetas de crédito, direcciones URL firmadas o información de identificación personal.

Cuotas para CodeCatalyst

En la siguiente tabla se describen las cuotas y los límites de Amazon CodeCatalyst. Puedes encontrar información adicional sobre aspectos específicos CodeCatalyst en los siguientes temas:

- [Cuotas para los repositorios de fuentes en CodeCatalyst](#)
- [Cuotas de identidad, permisos y acceso en CodeCatalyst](#)
- [Cuotas para flujos de trabajo en CodeCatalyst](#)
- [Cuotas para entornos de desarrollo en CodeCatalyst](#)
- [Cuotas para proyectos en CodeCatalyst](#)
- [Cuotas de planos en CodeCatalyst](#)
- [Cuotas de espacios en CodeCatalyst](#)
- [Cuotas para emisiones en CodeCatalyst](#)

Número máximo de espacios en una cuenta	5
Número máximo de espacios que un usuario puede crear en un mes natural	5
Número mínimo Cuentas de AWS de espacios	1
Número máximo de conexiones de cuentas para un espacio	5 000
Número máximo de conexiones de VPC para un espacio	100
Número máximo de proyectos en un espacio	100
Número máximo de proyectos a los que puede pertenecer un usuario	1 000
Descripciones de espacios	Las descripciones de los espacios son opcionales. Si se especifica, deben tener una longitud de entre 0 y 200 caracteres. Pueden contener cualquier combinación de letras,

números, espacios, puntos, guiones bajos, comas, guiones y los siguientes caracteres especiales:

? & \$ % + = / \ ; : \n \t \r

Nombres de espacios

Los nombres de los espacios deben ser únicos en todos ellos CodeCatalyst. No puede volver a utilizar los nombres de los espacios eliminados. Los nombres de los espacios deben tener una longitud de entre 3 y 63 caracteres. También deben empezar por un carácter alfanumérico. Los nombres de los espacios pueden contener cualquier combinación de letras, números, puntos, guiones bajos y guiones. No pueden contener ninguno de los siguientes caracteres:

! ? @ # \$ % ^ & * () + = { } []
| \ / > < ~ ` ' " ; :

Historial de documentos de Amazon CodeCatalyst

En la siguiente tabla se describe el historial de la documentación y las actualizaciones de la documentación general de CodeCatalyst.

Cambio	Descripción	Fecha
Contenido nuevo: uso de tareas para dividir los problemas en objetivos más pequeños	Se ha añadido contenido para facilitar el lanzamiento de tareas en los números. Se pueden añadir tareas a las cuestiones para desglosar las, organizarlas y realizar un seguimiento del trabajo de esas cuestiones.	4 de abril de 2024
Contenido actualizado: Limitaciones de GitHub las acciones en CodeCatalyst	Se actualizó el Limitaciones de GitHub las acciones en CodeCatalyst tema para indicar que GitHub las acciones se ejecutan en una imagen de Docker de un entorno de ejecución anterior.	2 de abril de 2024
Contenido actualizado: Trabajar con artefactos	Se actualizó el Trabajar con artefactos tema para incluir dos subtemas nuevos: y ¿Puedo compartir artefactos sin especificarlos como salidas y entradas? ¿Puedo compartir artefactos entre flujos de trabajo?	2 de abril de 2024
Contenido nuevo: referencia de acción «AWS CDK desplegar»	Se agregó una nueva <code>CloudAssemblyRootPath</code> propiedad referenci	1 de abril de 2024

	a de acción «AWS CDK desplegar» .	
Contenido actualizado: Trabajar con imágenes de Docker en un entorno de ejecución	Se actualizó el Trabajar con imágenes de Docker en un entorno de ejecución tema para incluir información sobre la nueva imagen del entorno de ejecución de marzo de 2024.	26 de marzo de 2024
Contenido actualizado: Trabajar con roles	Consolidó la información sobre los permisos de los roles en una sola tabla. La tabla está en un Permisos disponibles para cada función tema nuevo.	18 de marzo de 2024
Contenido nuevo: Ver todos los espacios y proyectos de un usuario	Se agregó información sobre la visualización de un listado en la página de inicio del usuario que muestra cada CodeCatalyst espacio o proyecto para el usuario que ha iniciado sesión. CodeCatalyst Consulte Ver todos los espacios y proyectos de un usuario .	18 de marzo de 2024
Contenido nuevo: Ejemplo: un gatillo que se tira y se ramifica	Se ha añadido un ejemplo de un activador de solicitudes de cambios. Hizo pequeñas correcciones en todo el Trabajar con factores desencadenantes tema.	11 de marzo de 2024

Contenido actualizado: trabajar con roles	Se actualizó la documentación de los roles para incluir los permisos para crear, eliminar y ver entornos.	4 de marzo de 2024
Contenido actualizado: Tutorial: Uso de funciones de IA generativa	Se ha actualizado el tutorial para reflejar los cambios a la hora de crear y asignar problemas a Amazon Q.	4 de marzo de 2024
Contenido nuevo: El bloqueo de las solicitudes de incorporación de cambios se fusiona cuando los flujos de trabajo GitHub fallan	Se ha añadido contenido nuevo sobre cómo bloquear una combinación de solicitudes de GitHub extracción con las reglas de protección de las sucursales cuando los flujos de trabajo fallan.	4 de marzo de 2024
Contenido nuevo: emite componentes	Se agregó contenido nuevo sobre cómo trabajar con los componentes de los problemas como desarrollador de planos personalizados.	27 de febrero de 2024
Contenido actualizado: Tipos de acción	Se ha actualizado el CodeCatalyst acciones tema para incluir una lista de las acciones de CodeCatalyst Labs.	21 de febrero de 2024
Contenido actualizado: funciona con solicitudes de incorporación de cambios	Se actualizó la documentación para reflejar la nueva funcionalidad con normas de aprobación y requisitos fundamentales para fusionar una solicitud de extracción.	15 de febrero de 2024

Contenido actualizado: Fusión de solicitudes de cambios	Se ha añadido documentación para las solicitudes de extracción, que incluye información sobre los requisitos de fusión imprescindibles para fusionar una solicitud de extracción que aún no ha recibido la aprobación de los revisores necesarios ni ha cumplido con las normas de aprobación.	15 de febrero de 2024
Contenido nuevo: gestiona las reglas de aprobación	Se ha añadido documentación para las solicitudes de cambios a fin de incluir información sobre la creación y la gestión de las reglas de aprobación.	15 de febrero de 2024
Contenido actualizado: trabajar con roles	Se actualizó la documentación de los roles para incluir los permisos para trabajar con las reglas de aprobación y las solicitudes de cambios.	14 de febrero de 2024
Contenido actualizado: ¿Cómo puedo corregir los errores «La definición de flujo de trabajo no tiene errores»?	Se actualizó la ¿Cómo puedo corregir los errores «La definición de flujo de trabajo no tiene errores»? sección para incluir más consejos de solución de problemas.	9 de febrero de 2024
Contenido nuevo: Visualización del estado del flujo de trabajo	Se agregó una sección que describe los estados del flujo de trabajo.	9 de febrero de 2024

Actualizar contenido: Cuotas para flujos de trabajo en CodeCatalyst	Se actualizó el Cuotas para flujos de trabajo en CodeCatalyst tema con el número máximo de acciones por flujo de trabajo y el número máximo de entornos asociados a una cuota Cuenta de AWS por espacio.	7 de febrero de 2024
Contenido actualizado: Creación de un entorno	Se actualizó la Creación de un entorno sección para indicar que se puede usar como máximo una conexión de cuenta por entorno.	31 de enero de 2024
Contenido nuevo: repositorio de planos GitHub personalizados	Se agregó contenido nuevo para el GitHub repositorio que está disponible públicamente.	10 de enero de 2024
Contenido actualizado: configurar npm con CodeCatalyst	Se actualizaron las instrucciones generales de configuración para usar npm con esta CodeCatalyst opción y se agregó claridad al respecto. <code>always-auth=true</code>	5 de enero de 2024
Contenido actualizado: funciona con solicitudes de cambios	Se actualizó la documentación para reflejar las nuevas funciones con las funciones generativas de IA incorporadas CodeCatalyst.	28 de noviembre de 2023
Contenido actualizado: creación de una edición	Se actualizó la documentación para reflejar las nuevas funciones con las funciones generativas de IA incorporadas CodeCatalyst.	28 de noviembre de 2023

Contenido nuevo: Tutorial: Uso de funciones de IA generativa	Se ha añadido un tutorial sobre el uso de las funciones de IA generativa en Amazon CodeCatalyst.	28 de noviembre de 2023
Contenido nuevo: planos personalizados y gestión del ciclo de vida	Se ha añadido contenido nuevo para utilizar las funciones personalizadas de gestión del ciclo de vida y los planos en Amazon CodeCatalyst.	27 de noviembre de 2023
Contenido actualizado: Tutorial: Creación de un proyecto con el modelo de aplicación web moderna de tres niveles	Se actualizó el tutorial con información sobre correcciones y solución de problemas.	22 de noviembre de 2023
Contenido actualizado: Trabajar con factores desencadenantes	Se corrigieron algunos ejemplos y descripciones relacionados con los activadores de las solicitudes de extracción. Se ha añadido una Consideraciones sobre los factores desencadenantes al ramificar sección.	22 de noviembre de 2023
Contenido nuevo: Iniciar sesión con SSO	Se agregó información sobre el inicio de sesión con el inicio de sesión único (SSO) y enlaces a información sobre cómo configurar y administrar un CodeCatalyst espacio que admita la federación de identidades. Consulte Configuración CodeCatalyst y Iniciar sesión con SSO .	17 de noviembre de 2023

Contenido actualizado: trabajar con roles	Se actualizó la documentación de los roles para incluir permisos para trabajar con equipos, conexiones de VPC, inicio de sesión único y recursos de máquinas.	16 de noviembre de 2023
Contenido actualizado: cómo trabajar con solicitudes de incorporación de cambios	Se actualizó la documentación para reflejar los cambios en la forma en que se muestran los cambios de una solicitud de extracción.	16 de noviembre de 2023
Contenido actualizado: Cuotas para CodeCatalyst	Se actualizó el Cuotas para CodeCatalyst tema con el número máximo de conexiones de VPC para una cuota de espacio.	16 de noviembre de 2023
Contenido nuevo: uso de entornos de desarrollo con una conexión de VPC	Se agregó documentación para usar entornos de desarrollo con una conexión de VPC.	16 de noviembre de 2023
Contenido nuevo: gestión de equipos para un espacio y para proyectos CodeCatalyst	Se agregó información sobre el uso de equipos con espacios. Consulte Gestión de equipos y Gestión de equipos para proyectos .	16 de noviembre de 2023
Contenido nuevo: administración de los recursos de la máquina para los planos y los flujos de trabajo en un espacio	Se agregó información sobre el uso de los recursos de la máquina con los espacios. Consulte Gestión de los recursos de la máquina .	16 de noviembre de 2023

Contenido nuevo: administración de los recursos de la máquina para los planos y los flujos de trabajo de un CodeCatalyst proyecto	Se agregó información sobre el uso de los recursos de la máquina en los CodeCatalyst proyectos. Consulte Gestión de los recursos de la máquina .	16 de noviembre de 2023
Contenido nuevo: asociación de una conexión de VPC a un entorno	Se agregó documentación para asociar una conexión de VPC a un entorno, que se puede usar en un flujo de trabajo.	16 de noviembre de 2023
Nuevo contenido	Publicación inicial de la Guía del CodeCatalyst administrador de Amazon .	16 de noviembre de 2023
Contenido nuevo: referencia de acción «AWS CDK desplegar»	Se agregó una nueva <code>CdkCliVersion</code> propiedad a la referencia de acción «AWS CDK desplegar» y a la referencia de acción «AWS CDKbootstrap» .	14 de noviembre de 2023
Contenido actualizado: Trabajar con roles	Se actualizó la documentación de los roles para incluir los permisos para trabajar con las reglas de las sucursales.	13 de noviembre de 2023
Contenido actualizado: solución de problemas con los repositorios de código fuente, los flujos de trabajo y los entornos de desarrollo	Se actualizaron los temas de solución de problemas para incluir información sobre cómo trabajar con las reglas de las sucursales.	13 de noviembre de 2023

Contenido actualizado: Referencia de acciones de compilación y prueba	Se actualizó la documentación de la Environment propiedad. Ahora es un campo opcional para las acciones de construcción y prueba.	13 de noviembre de 2023
Contenido nuevo: Gestiona las reglas de las sucursales	Se agregó documentación para las sucursales, que incluye información sobre la visualización de las reglas de las sucursales en un repositorio de origen y la creación y administración de las reglas de las sucursales.	13 de noviembre de 2023
Contenido actualizado: funciona con solicitudes de cambios	Se actualizó la documentación para reflejar los cambios en la forma en que se muestra la información sobre una solicitud de extracción.	10 de noviembre de 2023
Contenido actualizado: Trabajar con el almacenamiento en caché de archivos	Se actualizó la documentación para incluir las limitaciones de almacenamiento en caché de archivos.	10 de noviembre de 2023
Contenido actualizado: Tutorial: Implementación de una aplicación en Amazon EKS	Se actualizó la documentación para mencionar el plan de implementación de la aplicación EKS.	9 de noviembre de 2023
Contenido nuevo: Paquetes en CodeCatalyst	Se agregó documentación para usar paquetes en CodeCatalyst.	1 de noviembre de 2023

[Contenido nuevo y actualizado: trabajar con roles](#)

Se actualizó la documentación para cuatro nuevos roles CodeCatalyst: usuario avanzado, acceso limitado, revisor y solo lectura.

1 de noviembre de 2023

[Contenido actualizado: Trabajar con los parámetros de salida de GitHub Actions](#)

Se actualizaron los ejemplos para usar el archivo de GITHUB_OUTPUT entorno en lugar del set-output comando. El uso de archivos de entorno GitHub es el método recomendado para configurar los parámetros de salida.

24 de octubre de 2023

[Contenido nuevo: Trabajar con factores desencadenantes](#)

Se agregó documentación para programar los activadores.

16 de octubre de 2023

[Contenido actualizado: Referencia de acción «Implementar en un clúster de Kubernetes»](#)

Se agregó información sobre el uso del CodeCatalystWorkflowDevelopmentRole- *spaceName* rol a los [Tutorial: Implementación de una aplicación en Amazon EKS](#) temas [Referencia de acción «Implementar en un clúster de Kubernetes»](#) y.

22 de septiembre de 2023

[Contenido actualizado:
nuevo nombre y política
para el CodeCatalystWorkfl
owDevelopmentRole-
spaceName rol](#)

Se actualizaron los pasos y las descripciones de las funciones para cambiar el nombre de la función de desarrollador a CodeCatalystWorkflowDevelopmentRole-*spaceName* . El rol de desarrollador ahora usa la política AdministratorAccess AWS administrada. Consulte [Comprensión de la función CodeCatalystWorkflowDevelopmentRole-*spaceName* de servicio](#) y [Inscríbete para crear tu primer espacio y tu rol de desarrollo](#).

20 de septiembre de 2023

[Contenido actualizado:
Trabajar con variables](#)

Se introdujeron dos conceptos nuevos: variables definidas por el usuario y variables predefinidas. Estos conceptos deberían facilitar la lectura y la comprensión de la [Trabajar con variables](#) sección.

19 de septiembre de 2023

[Contenido actualizado:
trabajar con confirmaciones](#)

Se actualizó la documentación para reflejar el cambio en la información que se muestra y proporcionar detalles sobre cómo ver las confirmaciones con varios padres.

7 de septiembre de 2023

[Contenido nuevo: Trabajar con
factores desencadenantes](#)

Se agregó el siguiente ejemplo al [Trabajar con factores desencadenantes](#) tema: [Ejemplo: un simple disparador de «pulsar el botón principal»](#)

6 de septiembre de 2023

Contenido actualizado: funciona con solicitudes de cambios	Se actualizó la documentación para reflejar el cambio en el orden de visualización de la sucursal de origen y la sucursal de destino al crear una solicitud de extracción.	30 de agosto de 2023
Contenido nuevo: consulta y cambia la rama predeterminada	Se agregó documentación para las sucursales para incluir información sobre cómo ver y cambiar la rama predeterminada de un repositorio de origen.	30 de agosto de 2023
Contenido actualizado: Referencia de acción «Implementar en un clúster de Kubernetes»	Se agregó una nota sobre Helm y Kustomize a la descripción de la Manifests propiedad en el. Referencia de acción «Implementar en un clúster de Kubernetes»	15 de agosto de 2023
Contenido nuevo: Gestión de los archivos adjuntos de los números	Se ha añadido documentación para trabajar con los archivos adjuntos relacionados con los problemas y gestionarlos.	15 de agosto de 2023
Contenido actualizado: Trabajar con factores desencadenantes	Se mejoró y amplió la documentación relacionada con los activadores del flujo de trabajo.	11 de agosto de 2023

[Contenido nuevo: solución de problemas de permisos de rol](#)

Se ha añadido información sobre la actualización de los permisos del rol para ejecutar un flujo de trabajo que requiere acceso a Amazon CodeGuru. Consulte [El flujo de trabajo moderno del blueprint de aplicaciones web de tres niveles OnPullRequest falla debido a un error de permisos para Amazon CodeGuru](#).

11 de agosto de 2023

[Contenido nuevo: ¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?](#)

Se agregó el siguiente tema de solución de problemas : [¿Cómo puedo corregir los mensajes que dicen que el flujo de trabajo está inactivo?](#)

11 de agosto de 2023

[Contenido nuevo: Añadir la acción «Implementar en un clúster de Kubernetes»](#)

Se agregó documentación para la acción Implementar en un clúster de Kubernetes. Para obtener más información, consulte [Añadir la acción «Implementar en un clúster de Kubernetes»](#) y [Tutorial: Implementación de una aplicación en Amazon EKS](#).

27 de julio de 2023

[Actualizaciones sobre la forma en que se registran los eventos de administración de un espacio CodeCatalyst](#)

Se agregó información sobre cómo se registran los eventos de administración para acciones específicas en un CodeCatalyst espacio con AWS CloudTrail. Se agregó información sobre cómo se pueden ver todos los eventos de un espacio con el list-events-logs comando. Consulte [Monitorización en Amazon CodeCatalyst](#).

20 de julio de 2023

[Contenido actualizado: Trabajar con factores desencadenantes](#)

Se actualizó la documentación para indicar que los activadores de solicitudes de incorporación de cambios ahora son compatibles con los repositorios de GitHub origen. Anteriormente, los activadores de solicitudes de extracción solo se admitían en los repositorios CodeCatalyst de origen.

14 de julio de 2023

[Contenido actualizado: Referencia de definición de flujo de trabajo](#)

Se ha corregido un error de formato en el bloque Compute de código.

27 de junio de 2023

[Contenido actualizado: Cuotas para flujos de trabajo en CodeCatalyst](#)

Se ha actualizado el [Cuotas para flujos de trabajo en CodeCatalyst](#) tema con la cuota máxima de tiempo que puede ejecutarse una acción.

27 de junio de 2023

Contenido actualizado: Protección de datos	Se actualizó la documentación para incluir información adicional sobre la replicación de datos.	26 de junio de 2023
Contenido nuevo: Trabajar con versiones de acción	Se ha añadido un Trabajar con versiones de acción tema.	21 de junio de 2023
Contenido actualizado: Trabajo con entornos	Se ha aclarado la ¿Qué acciones son compatibles con los entornos? sección.	14 de junio de 2023
Contenido actualizado: documentación de problemas reorganizada	Se reorganizó la mayoría de la documentación de problemas para alinearla mejor con el conjunto general de documentación y los flujos de usuarios.	31 de mayo de 2023
Contenido actualizado: conmutador de vistas de problemas	Se actualizaron varios flujos de usuarios para alinearlos con el conmutador de visualización de problemas actualizado.	31 de mayo de 2023
Contenido actualizado: gestión de las notificaciones	Se actualizó la documentación sobre las notificaciones para incluir información sobre la configuración de las notificaciones personales de Slack.	30 de mayo de 2023
Contenido actualizado: Gestión de las notificaciones	Se actualizó la documentación sobre las notificaciones para incluir información sobre la configuración de las notificaciones personales de Slack.	30 de mayo de 2023

Contenido nuevo: modelo de CodeCatalyst confianza	Se agregó un nuevo tema con información sobre el modelo de confianza, que CodeCatalyst permite asumir el rol de servicio en los conectados Cuenta de AWS. Se agregó una nueva sección sobre los principios de servicio definidos para CodeCatalyst. Consulte CodeCatalyst modelo de confianza .	20 de mayo de 2023
Contenido actualizado: Trabajar con fuentes	Simplificó las instrucciones en Hacer referencia a archivos de un repositorio fuente .	10 de mayo de 2023
Contenido actualizado: Cuotas para flujos de trabajo en CodeCatalyst	Se actualizó el Cuotas para flujos de trabajo en CodeCatalyst tema con la longitud máxima de una cuota de valor variable de salida.	10 de mayo de 2023
Contenido nuevo: Trabajar con artefactos	Se agregaron dos ejemplos: Ejemplo: hacer referencia a un archivo en un único artefacto y Ejemplo: hacer referencia a un archivo en un artefacto cuando hay un WorkflowSource .	10 de mayo de 2023
Contenido actualizado: funciona con solicitudes de cambios	Se actualizó la documentación de las solicitudes de extracción para incluir información sobre la configuración de las preferencias de correo electrónico para los eventos de solicitudes de extracción.	21 de abril de 2023

Contenido actualizado: Administración de notificaciones	Se actualizó la documentación de las notificaciones para incluir información sobre la configuración de las preferencias de correo electrónico para los eventos de solicitudes de extracción de cambios.	21 de abril de 2023
Actualizaciones de las políticas administradas	Se agregaron Política administrada de AWS: AmazonCodeCatalyst FullAccess las Política administrada de AWS: AmazonCodeCatalyst ReadOnlyAccess políticas y se Política administrada de AWS: AmazonCodeCatalyst SupportAccess administraron. Consulte Actualizaciones de CodeCatalyst en las políticas administradas de AWS .	20 de abril de 2023
Contenido nuevo: Eliminar un objetivo de despliegue	Se ha añadido un Eliminar un objetivo de despliegue tema.	20 de abril de 2023
Contenido nuevo: Tipos de acción	Se ha añadido un CodeCatalyst Acciones de Labs tema.	20 de abril de 2023
Actualizaciones para administrar un usuario con el rol de administrador de espacios en un espacio	Se agregó información sobre cómo quitar o cambiar el rol de un usuario con el rol de administrador de espacios en un espacio. Consulte Eliminar o cambiar el rol de un usuario con el rol de administrador de Space .	19 de abril de 2023

Actualizaciones para administrar entornos de desarrollo	Se agregó información sobre la administración de entornos de desarrollo como administrador de Space. Consulte Administración de entornos de desarrollo para un espacio .	19 de abril de 2023
Contenido actualizado: problemas de búsqueda y visualización	Reorganizó el Problemas de búsqueda y visualización tema y los subtemas.	19 de abril de 2023
Contenido actualizado: Trabajar con computación	Se agregó soporte para la arquitectura Arm64 en Amazon Linux 2.	19 de abril de 2023
Contenido nuevo: mover los problemas dentro de los grupos	Se ha añadido documentación para trasladar las cuestiones de un grupo a la Junta y a las vistas «Todos los temas».	19 de abril de 2023
Contenido actualizado: Cuotas para flujos de trabajo en CodeCatalyst	Se actualizó el Cuotas para flujos de trabajo en CodeCatalyst tema en el que faltaban cuotas y se actualizó la cuota del tamaño total máximo de las variables de salida de una sola acción a 120 KB (en lugar de 2 KB).	18 de abril de 2023
Contenido nuevo: Visualización del código fuente de una acción	Se ha añadido un Visualización del código fuente de una acción tema.	18 de abril de 2023
Contenido nuevo: Reintentar casos de prueba	Se ha añadido un Reintentar casos de prueba tema.	11 de abril de 2023

Contenido nuevo: Detener una ejecución de flujo de trabajo	Se ha añadido un Detener una ejecución de flujo de trabajo tema.	10 de abril de 2023
Contenido nuevo: se agregaron secciones para etiquetar los recursos para las conexiones de cuentas entre AWS y Amazon CodeCatalyst	Se agregó información para etiquetar los recursos de conexión de cuentas y administrar las políticas de IAM para los recursos de conexión. Consulte Uso de etiquetas para controlar el acceso a los recursos de conexión de la cuenta y CodeCatalyst referencia de permisos .	6 de abril de 2023
Contenido nuevo: Tipos de acción	Se ha añadido un Tipos de acción tema.	6 de abril de 2023
Contenido actualizado: Referencia de acción «Implementar AWS CloudFormation pila»	Se ha actualizado la descripción <code>parameter-override</code> de la propiedad. Ahora es compatible con archivos JSON.	5 de abril de 2023
Contenido nuevo: creación de un proyecto CodeCatalyst con un GitHub repositorio vinculado	Se agregó una nueva sección Crear un proyecto en Amazon CodeCatalyst titulada Crear un proyecto con un GitHub repositorio vinculado con instrucciones para crear un proyecto que enlace a tu GitHub repositorio.	5 de abril de 2023

Contenido actualizado: funciona con notificaciones	Se actualizó la documentación de las notificaciones para incluir información sobre la configuración de los correos electrónicos relacionados con los eventos del proyecto.	31 de marzo de 2023
Nuevo contenido	Publicación inicial de la guía Amazon CodeCatalyst Action Development Kit .	31 de marzo de 2023
Contenido actualizado: se ha reestructurado la sección de espacios en Amazon CodeCatalyst	Se actualizó la sección de espacios eliminando las páginas de destino y consolidando los temas.	29 de marzo de 2023
Contenido actualizado: Tutorial: Implementación de una aplicación en Amazon ECS	Se modificó Paso 1: Configura un AWS usuario y AWS CloudShell para describir cómo crear un usuario en AWS IAM Identity Center lugar de AWS Identity and Access Management. Ya no se recomienda crear usuarios de IAM.	23 de marzo de 2023
Contenido actualizado: trabajar con roles	Se actualizó la documentación de las funciones de administrador del espacio, administrador de proyectos y colaborador para incluir los permisos necesarios para vincular las incidencias con las solicitudes de incorporación de cambios.	13 de marzo de 2023

Contenido actualizado: se trabaja con solicitudes de incorporación de cambios	Se actualizó la documentación de las solicitudes de extracción para incluir información sobre cómo vincular los problemas con las solicitudes de extracción.	13 de marzo de 2023
Contenido actualizado: Cómo solucionar problemas	Se actualizó la documentación sobre los problemas para incluir información sobre cómo vincular los problemas a las solicitudes de cambios.	13 de marzo de 2023
Contenido nuevo: Ver el estado y los detalles de todas las ejecuciones del proyecto	Se agregó una sección que describe la nueva página de ejecución del flujo de trabajo agregado.	8 de marzo de 2023
Contenido nuevo: ¿Cómo puedo corregir los errores «La definición de flujo de trabajo no tiene errores»?	Se agregó una sección sobre cómo solucionar los errores «La definición del flujo de trabajo tiene errores».	7 de marzo de 2023
Contenido actualizado: Crear, editar y eliminar un flujo de trabajo	Se actualizaron las instrucciones para reflejar la nueva interfaz de usuario.	3 de marzo de 2023
Contenido nuevo: Uso universal-test-runner	Se ha añadido un Uso universal-test-runner tema.	3 de marzo de 2023
Contenido actualizado: Cree, pruebe e implemente con flujos de trabajo en CodeCatalyst	Se actualizaron varias secciones para reflejar los nuevos filtros de nombres de repositorios, ramas y flujos de trabajo de origen en la página de resumen de los flujos de trabajo.	2 de marzo de 2023

Contenido nuevo: Visualización de la calidad del código y el estado de la implementación por confirmación CodeCatalyst	Se agregó una sección sobre la visualización de la calidad del código y el estado de la implementación por confirmación.	27 de febrero de 2023
Contenido nuevo: variables BranchName «» y CommitId «»	Se ha añadido una nueva variable BranchName predefinida.	16 de febrero de 2023
Contenido actualizado: Gestión de los miembros del espacio en Amazon CodeCatalyst	Se actualizó la información sobre el cambio de los roles de los miembros, la invitación y la eliminación de miembros en dos tablas nuevas en función del rol asignado al usuario en CodeCatalyst.	15 de febrero de 2023
Contenido actualizado: se agregaron pasos para la administración de PAT en la CodeCatalyst consola de Amazon	Se agregaron pasos para ver, crear y eliminar las PAT en la consola.	15 de febrero de 2023
Contenido actualizado: Trabajar con imágenes de Docker en un entorno de ejecución	Se agregaron más herramientas a la tabla de versiones de las herramientas de imagen predeterminadas.	10 de enero de 2023
Contenido actualizado: Trabajar con artefactos	Se ha corregido la trayectoria de un artefacto.	3 de enero de 2023
Contenido actualizado: Trabajar con fuentes	Se ha corregido una ruta de origen.	3 de enero de 2023
Contenido actualizado: Referencia de acción «GitHub Acciones»	Se ha corregido el fragmento de código de la Steps sección.	3 de enero de 2023

Contenido actualizado: Actualización de una solicitud de cambios	Se actualizó la documentación para incluir información sobre la actualización de los revisores obligatorios u opcionales de una solicitud de extracción.	23 de diciembre de 2022
Contenido nuevo: Trabajar con el almacenamiento en caché de archivos	Se agregó una página para almacenar archivos en caché en un flujo de trabajo.	20 de diciembre de 2022
Contenido actualizado: funciona con solicitudes de cambios	Se actualizó la documentación de las solicitudes de cambios para incluir información sobre las notificaciones.	16 de diciembre de 2022
Contenido nuevo: referencia de acción «AWS CDK desplegar»	Se ha añadido una nueva <code>CdkRootPath</code> propiedad.	16 de diciembre de 2022
Contenido nuevo: Compartir el cómputo entre las acciones	Se ha añadido un Compartir el cómputo entre las acciones tema.	14 de diciembre de 2022
Contenido actualizado: Trabajar con artefactos	Se corrigieron los ejemplos que mostraban cómo especificar los artefactos de entrada.	13 de diciembre de 2022
Contenido nuevo: Referencia de acción «GitHub Acciones»	Se ha añadido una página de referencia dedicada a la acción GitHub Acciones.	13 de diciembre de 2022
Contenido actualizado: Cuotas para proyectos en CodeCatalyst	Se actualizó la documentación con un máximo de 100 proyectos en un espacio.	2 de diciembre de 2022

[Nuevo contenido](#)

Publicación inicial de la Guía del CodeCatalyst usuario de Amazon.

1 de diciembre de 2022

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el [AWS glosario](#) de la Glosario de AWS Referencia.

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