



Guía del usuario de

AWS CodeCommit



Versión de API 2015-04-13

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

AWS CodeCommit: Guía del usuario de

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

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

Table of Contents

¿Qué es CodeCommit?	1
Presentamos CodeCommit	2
CodeCommit, Git y elegir el AWS servicio adecuado para sus necesidades	3
¿Cómo CodeCommit funciona?	7
¿En qué se CodeCommit diferencia del control de versiones de archivos en Amazon S3?	9
¿Cómo puedo empezar CodeCommit?	9
¿Dónde puedo encontrar más información sobre Git?	9
Configuración	11
Visualización y administración de sus credenciales	11
Configuración mediante credenciales de Git	12
Configuración con otros métodos	12
Compatibilidad entre CodeCommit, Git y otros componentes	14
Para usuarios HTTPS mediante credenciales de Git	15
Paso 1: Configuración inicial para CodeCommit	16
Paso 2: Instalar Git	17
Paso 3: Crear credenciales de Git para las conexiones HTTPS a CodeCommit	17
Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio	19
Siguientes pasos	21
Para conexiones HTTPS con git-remote-codecommit	21
Paso 0: Instalar los requisitos previos para git-remote-codecommit	22
Paso 1: Configuración inicial para CodeCommit	23
Paso 2: Instalar git-remote-codecommit	27
Paso 3: Conectarse a la CodeCommit consola y clonar el repositorio	28
Siguientes pasos	29
Para conexiones desde herramientas de desarrollo	29
Integre AWS Cloud9 con AWS CodeCommit	32
Integrar Visual Studio con AWS CodeCommit	37
Para los usuarios de SSH que no utilizan AWS CLI	38
Paso 1: Asociar su clave pública a su usuario de IAM	39
Paso 2: CodeCommit Añádalo a tu configuración de SSH	40
Siguientes pasos	40
Para conexiones SSH en Linux, macOS o Unix	41
Paso 1: Configuración inicial para CodeCommit	41
Paso 2: Instalar Git	42

Paso 3: Configurar las credenciales en Linux, macOS o Unix	43
Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio	47
Siguientes pasos	49
Para conexiones SSH en Windows	49
Paso 1: Configuración inicial para CodeCommit	49
Paso 2: Instalar Git	50
Paso 3: Configura las claves públicas y privadas para Git y CodeCommit	51
Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio	55
Siguientes pasos	57
Para conexiones HTTPS en Linux, macOS o Unix con el asistente de AWS CLI credenciales	57
Paso 1: Configuración inicial para CodeCommit	58
Paso 2: Instalar Git	61
Paso 3: Configurar el ayudante de credenciales	62
Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio	64
Siguientes pasos	65
Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI	65
Paso 1: Configuración inicial para CodeCommit	66
Paso 2: Instalar Git	70
Paso 3: Configurar el ayudante de credenciales	71
Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio	73
Siguientes pasos	74
Introducción	75
Empezando con CodeCommit	75
Requisitos previos	76
Paso 1: Crear un CodeCommit repositorio	77
Paso 2: Añadir archivos a su repositorio	80
Paso 3: Examinar el contenido del repositorio	82
Paso 4: Crear y colaborar en una solicitud de extracción	87
Paso 5: Eliminar	94
Paso 6: Siguiendo pasos	95
Primeros pasos con Git y CodeCommit	95
Paso 1: Crear un CodeCommit repositorio	97
Paso 2: Crear un repositorio local	98
Paso 3: Crear la primera confirmación	100
Paso 4: Enviar la primera confirmación	101
Paso 5: Comparte el CodeCommit repositorio y envía y extrae otra confirmación	102

Paso 6: Crear y compartir una ramificación	104
Paso 7: Crear y compartir una etiqueta	106
Paso 8: Configurar los permisos de acceso	107
Paso 9: limpiar	111
Integraciones de productos y servicios	113
Integración con otros servicios de AWS	113
Ejemplos de integración de la comunidad	121
Publicaciones de blog	122
Ejemplos de código	125
Trabajar con repositorios	127
Creación de un repositorio	128
Crear un repositorio (Consola)	129
Crea un repositorio (AWS CLI)	131
Conectarse a un repositorio	133
Requisitos previos para conectarse a un repositorio CodeCommit	134
Conéctese al CodeCommit repositorio clonando el repositorio	135
Conectar un repositorio local al CodeCommit repositorio	136
Compartir un repositorio	138
Seleccione el protocolo de conexión para compartir con sus usuarios	139
Creación de políticas de IAM para su repositorio	140
Creación de un grupo IAM para usuarios de repositorios	142
Compartir la información de conexión con los usuarios	143
Configuración de notificaciones para eventos del repositorio	144
Uso de reglas de notificación de repositorios	146
Creación de una regla de notificación	146
Cambiar o deshabilitar notificaciones	150
Eliminación de notificaciones	151
Etiquetado de un repositorio	152
Añadir una etiqueta a un repositorio	153
Visualización de etiquetas para un repositorio	155
Editar etiquetas de un repositorio	157
Eliminar una etiqueta de un repositorio	159
Administración de los disparadores de un repositorio	160
Cree el recurso y añada permisos para CodeCommit	161
Creación de un disparador para un tema de Amazon SNS	162
Creación de un disparador para una función de Lambda	169

Crear un disparador para una función de Lambda existente	174
Edición de los disparadores de un repositorio	182
Prueba de disparadores para un repositorio	184
Eliminación de disparadores de un repositorio	187
Amazon CodeGuru Reviewer y CodeCommit	189
Visualización de la información de los repositorios	190
Visualización de la información de los repositorios (consola)	190
Ver detalles CodeCommit del repositorio (Git)	191
Ver los detalles del CodeCommit repositorio (AWS CLI)	193
Cambio de la configuración de un repositorio	196
Cambio de la configuración de un repositorio (consola)	197
Cambio AWS CodeCommit configuración del repositorio (AWS CLI)	199
Sincronización de los cambios entre repositorios	201
Envío de confirmaciones a dos repositorios	202
Configuración del acceso entre cuentas a un repositorio mediante roles	207
Cross-account acceso al repositorio: acciones para el administrador en AccountA	208
Cross-account acceso al repositorio: acciones para el administrador en AccountB	212
Cross-account acceso al repositorio: acciones para el usuario del repositorio en AccountB .	214
Eliminar un repositorio	220
Eliminar un CodeCommit repositorio (consola)	220
Eliminar un repositorio local	221
Eliminar un CodeCommit repositorio (AWS CLI)	221
Trabajar con archivos	223
Navegación por los archivos en un repositorio	224
Navegue por un CodeCommit repositorio	225
Crear o añadir un archivo	226
Crear o cargar un archivo (Consola)	226
Añadir un archivo (AWS CLI)	228
Añadir de un archivo (Git)	229
Edición del contenido de un archivo	229
Editar un archivo (consola)	230
Editar o eliminar un archivo (AWS CLI)	231
Editar un archivo (Git)	233
Trabajo con solicitudes de extracción	234
Crear una solicitud de extracción	238
Crear una solicitud de extracción (Consola)	238

Crear una solicitud de extracción (AWS CLI)	240
Crear una regla de aprobación	241
Crear una regla de aprobación para una solicitud de extracción (consola)	242
Crear una regla de aprobación para una solicitud de extracción (AWS CLI)	245
Visualización de solicitudes de extracción	246
Visualizar solicitudes de extracción (consola)	246
Visualizar solicitudes de extracción (AWS CLI)	247
Revisión de una solicitud de extracción	252
Revisión de una solicitud de extracción (consola)	252
Revisar solicitudes de extracción (AWS CLI)	258
Actualización de una solicitud de extracción	263
Actualizar una solicitud de extracción (consola)	263
Actualizar solicitudes de extracción (AWS CLI)	264
Editar o eliminar una regla de aprobación	265
Editar o eliminar una regla de aprobación para una solicitud de extracción (consola)	266
Editar o eliminar una regla de aprobación para una solicitud de extracción (AWS CLI)	268
Anular reglas de aprobación en una solicitud de extracción	269
Anular reglas de aprobación (consola)	270
Anular reglas de aprobación (AWS CLI)	270
Combinación de una solicitud de extracción	272
Combinación de una solicitud de extracción (consola)	273
Combinar una solicitud de extracción (AWS CLI)	276
Resolver conflictos en una solicitud de extracción	283
Resolución de conflictos en una solicitud de extracción (consola)	283
Resolución de conflictos en una solicitud de extracción (AWS CLI)	286
Cierre de una solicitud de extracción	295
Cierre de una solicitud de extracción (consola)	295
Cerrar una solicitud de extracción (AWS CLI)	296
Trabajar con plantillas de reglas de aprobación	298
Crear una plantilla de regla de aprobación	300
Crear una plantilla de regla de aprobación (consola)	300
Crear una plantilla de regla de aprobación (AWS CLI)	305
Asociar una plantilla de regla de aprobación con un repositorio	307
Asociar una plantilla de regla de aprobación (consola)	307
Asociar una plantilla de regla de aprobación (AWS CLI)	308
Administrar plantillas de reglas de aprobación	309

Administrar plantillas de reglas de aprobación (consola)	309
Administrar plantillas de reglas de aprobación (AWS CLI)	310
Desasociar una plantilla de regla de aprobación	314
Desasociar una plantilla de regla de aprobación (consola)	315
Desasociar una plantilla de regla de aprobación (AWS CLI)	315
Eliminar una plantilla de regla de aprobación	316
Eliminar una plantilla de regla de aprobación (consola)	317
Eliminar una plantilla de regla de aprobación (AWS CLI)	317
Uso de confirmaciones	319
Creación de una confirmación	320
Cree la primera confirmación para un repositorio mediante el AWS CLI	321
Creación de una confirmación mediante un cliente de Git	322
Crea una confirmación mediante el AWS CLI	326
Ver detalles de una confirmación	329
Examinación de confirmaciones de un repositorio	329
Ver los detalles de la confirmación (AWS CLI)	333
Visualización de la información detallada de una confirmación (Git)	339
Comparar confirmaciones	342
Comparación de una confirmación con su elemento principal	342
Comparación de dos especificadores de confirmación	345
Comentar una confirmación	348
Ver los comentarios de una confirmación en un repositorio	348
Añadir y responder a los comentarios de una confirmación en un repositorio	349
Ver, añadir, actualizar y responder a los comentarios (AWS CLI)	354
Crear una etiqueta de Git	363
Uso de Git para crear una etiqueta	363
Visualización de información detallada de las etiquetas	364
Ver detalles de las etiquetas (consola)	365
Ver detalles de las etiquetas de Git (Git)	366
Eliminar una etiqueta	368
Uso de Git para eliminar una etiqueta de Git	368
Uso de las ramificaciones	370
Creación de una ramificación	372
Creación de una ramificación (consola)	372
Creación de una ramificación (Git)	373
Crea una rama (AWS CLI)	374

Limitación de combinaciones e inserciones en ramificaciones	376
Configuración de una política de IAM para limitar las inserciones y combinaciones en una ramificación	377
Aplicación de una política de IAM a un grupo o rol de IAM	379
Comprobación de la política	380
Visualización de la información detallada de la ramificación	380
Ver detalles de las ramificaciones (consola)	381
Visualización de la información detallada de la ramificación (Git)	382
Ver detalles de la sucursal (AWS CLI)	382
Comparación y combinación de ramificaciones	384
Comparación de una ramificación y la ramificación predeterminada	384
Comparación de dos ramificaciones específicas	385
Combina dos ramas (AWS CLI)	386
Cambiar la configuración de una ramificación	389
Cambiar la ramificación predeterminada (consola)	389
Cambie la rama predeterminada (AWS CLI)	390
Eliminación de una ramificación	391
Eliminación de una ramificación (consola)	392
Eliminar una rama (AWS CLI)	392
Eliminación de una ramificación (Git)	393
Trabajo con las preferencias del usuario	395
Migrar a CodeCommit	396
Migrar un repositorio de Git a AWS CodeCommit	396
Paso 0: Se requiere una configuración para acceder a CodeCommit	397
Paso 1: Crea un repositorio CodeCommit	403
Paso 2: clona el repositorio y envíalo al CodeCommit repositorio	405
Paso 3: Ver los archivos en CodeCommit	407
Paso 4: Compartir el repositorio CodeCommit	407
Migre el contenido a CodeCommit	410
Paso 0: Se requiere una configuración para acceder a CodeCommit	411
Paso 1: Crea un repositorio CodeCommit	417
Paso 2: migra el contenido local al CodeCommit repositorio	418
Paso 3: Ver los archivos en CodeCommit	420
Paso 4: Comparte el CodeCommit repositorio	420
Migrar de forma incremental un repositorio	423
Paso 0: Determinar si realizar una migración de forma incremental	423

Paso 1: Instale los requisitos previos y añada el CodeCommit repositorio como remoto	424
Paso 2: Crear el script para migrar de forma incremental	426
Paso 3: Ejecute el script y migre de forma incremental a CodeCommit	427
Anexo: script de muestra incremental-repo-migration.py	428
Seguridad	437
Protección de los datos	437
AWS KMS y cifrado	438
Uso de credenciales rotatorias	441
Gestión de identidad y acceso	446
Público	446
Autenticación con identidades	447
Administración del acceso con políticas	448
Autenticación y control de acceso	450
¿Cómo AWS CodeCommit funciona con IAM	518
CodeCommit políticas basadas en recursos	519
Autorización basada en etiquetas CodeCommit	519
CodeCommit Funciones de IAM	522
Ejemplos de políticas basadas en identidades	523
Resolución de problemas	526
Resiliencia	529
Seguridad de la infraestructura	529
Supervisión de CodeCommit	531
Supervisión de eventos de CodeCommit	532
Evento referenceCreated	533
Evento referenceUpdated	534
Evento referenceDeleted	534
Evento unreferencedMergeCommitCreated	535
Evento commentOnCommitCreated	536
Evento commentOnCommitUpdated	537
Evento commentOnPullRequestCreated	538
Evento commentOnPullRequestUpdated	539
Evento pullRequestCreated	539
Evento pullRequestSourceBranchUpdated	541
Evento pullRequestStatusChanged	542
Evento pullRequestMergeStatusUpdated	543
Evento approvalRuleTemplateCreated	544

Evento approvalRuleTemplateUpdated	544
Evento approvalRuleTemplateDeleted	545
Evento approvalRuleTemplateAssociatedWithRepository	546
Evento approvalRuleTemplateDisassociatedWithRepository	547
Evento approvalRuleTemplateBatchAssociatedWithRepositories	548
Evento approvalRuleTemplateBatchDisassociatedFromRepositories	549
Evento pullRequestApprovalRuleCreated	549
Evento pullRequestApprovalRuleDeleted	551
Evento pullRequestApprovalRuleOverridden	552
Evento pullRequestApprovalStateChanged	554
Evento pullRequestApprovalRuleUpdated	556
Evento reactionCreated	557
Evento reactionUpdated	558
Registrar llamadas a la AWS CodeCommit API con AWS CloudTrail	559
CodeCommit información en CloudTrail	559
Descripción de las entradas de los archivos de CodeCommit registro	561
AWS CloudFormation recursos	569
CodeCommit y CloudFormation plantillas	569
Ejemplos de plantillas	570
CloudFormation, CodeCommit, y el AWS Cloud Development Kit (AWS CDK)	571
Obtenga más información sobre CloudFormation	572
Solución de problemas	573
Solución de problemas de credenciales de Git (HTTPS)	573
Credenciales de Git para AWS CodeCommit: sigo viendo una solicitud de credenciales cuando me conecto a mi CodeCommit repositorio en la terminal o en la línea de comandos	574
Credenciales de Git para AWS CodeCommit: configuré las credenciales de Git, pero mi sistema no las usa	574
Solución de problemas de git-remote-codecommit	575
Veo un error: git: 'remote-codecommit' no es un comando de git	575
Aparece un error: error grave: no se ha podido encontrar el asistente remoto para 'codecommit'	576
Error de clonación: no puedo clonar un CodeCommit repositorio desde un IDE	576
Error de inserción o extracción: no puedo enviar o extraer confirmaciones de un IDE a un repositorio CodeCommit	576

Error de instalación: aparece un externally-managed-environment error al intentar realizar la instalación git-remote-codecommit	577
Solución de problemas de conexiones SSH	577
Error de acceso: la clave pública se ha cargado correctamente en IAM pero se ha producido un error de conexión en los sistemas Linux, macOS o Unix	578
Error de acceso: la clave pública se carga correctamente en IAM y SSH se prueba correctamente pero la conexión falla en los sistemas Windows	579
Problema de autenticación: no se puede establecer la autenticidad del host al conectarse a un repositorio CodeCommit	580
Error de IAM: "Formato no válido" al intentar añadir una clave pública en IAM	587
Necesito acceder a los CodeCommit repositorios de varias cuentas de Amazon Web Services con credenciales SSH	587
Git en Windows: el emulador de Bash o la línea de comandos se congela al intentar conectar mediante SSH	588
El formato de clave pública requiere una especificación en algunas distribuciones de Linux	589
Error de acceso: se denegó la clave pública SSH al conectarse a un repositorio CodeCommit	589
Solución de problemas del ayudante de credenciales (HTTPS)	590
Recibo un error al ejecutar el comando <code>git config</code> para configurar el ayudante de credenciales	590
Aparece un error al leer el nombre de usuario cuando intento clonar un repositorio	591
Recibo un error de comando no encontrado en Windows al usar el ayudante de credenciales	591
Se me pide un nombre de usuario cuando me conecto a un repositorio CodeCommit	592
Git para macOS: he configurado el ayudante de credenciales correctamente, pero ahora no tengo acceso a mi repositorio (403)	592
Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio (403)	596
Solución de problemas de clientes de Git	598
Error de Git: error: error de RPC; resultado = 56, código HTTP = 200 grave: El extremo remoto se ha interrumpido de forma inesperada	599
Error de Git: demasiados comandos de actualización de referencia	599
Error de Git: envío a través de HTTPS dividido en algunas versiones de Git	599
Error de Git: 'gnutls_handshake() failed'	600

Error de Git: Git no encuentra el CodeCommit repositorio o no tiene permiso para acceder al repositorio	600
Git en Windows: métodos de autenticación no admitidos disponibles (clave pública)	600
Solución de errores de acceso	601
Error de acceso: se me pide un nombre de usuario y una contraseña cuando me conecto a un CodeCommit repositorio desde Windows	601
Error de acceso: clave pública denegada al conectarse a un repositorio CodeCommit	602
Error de acceso: aparece el mensaje «Se ha superado la velocidad» o «429» al conectarse a un repositorio CodeCommit	602
Solución de errores de configuración	603
Error de configuración: no se pueden configurar AWS CLI las credenciales en macOS	604
Solución de problemas de errores de la consola	604
Error de acceso: se denegó el acceso a la clave de cifrado de un CodeCommit repositorio desde la consola o AWS CLI	602
Error de cifrado: no se puede descifrar el repositorio	605
Error de consola: no se puede examinar el código de un CodeCommit repositorio desde la consola	605
Error de visualización: no se puede ver un archivo o una comparación entre archivos	606
Solución de problemas de disparadores	606
Error de disparador: un disparador del repositorio no se ejecuta cuando se espera	606
Activar la depuración	607
CodeCommit referencia	609
Regiones y puntos de conexión de Git	609
compatible Regiones de AWS para CodeCommit	610
Puntos de conexión de Git	611
Huellas digitales del servidor para CodeCommit	618
Utilización AWS CodeCommit con puntos finales de VPC de interfaz	625
Disponibilidad.	626
Cree puntos finales de VPC para CodeCommit	627
Cree una política de puntos de conexión de VPC para CodeCommit	628
Cuotas	629
Referencia de la línea de comandos	637
Comandos básicos de Git	643
Variables de configuración	643
Repositorios remotos	644
Confirmaciones	646

Ramificaciones	647
Tags	649
Historial de revisión	651
Actualizaciones anteriores	663
AWSGlosario de	671
.....	dclxxii

¿Qué es AWS CodeCommit?

AWS CodeCommit es un servicio de control de versiones hospedado por Amazon Web Services que puede utilizar para almacenar y gestionar de forma privada activos (como documentos, código fuente y archivos binarios) en la nube. Para obtener información sobre los precios de CodeCommit, consulte [Precios](#).

Note

CodeCommit está dentro del ámbito de aplicación de muchos programas de conformidad. Para obtener información detallada AWS y sobre las iniciativas de cumplimiento, consulte [AWS Servicios incluidos en el programa de cumplimiento](#).

Se trata de un servicio compatible con HIPAA. [Para obtener más información sobre AWS la Ley de Portabilidad y Responsabilidad de los Seguros de Salud de los Estados Unidos de 1996 \(HIPAA\) y el uso de AWS los servicios para procesar, almacenar y transmitir información de salud protegida \(PHI\), consulte Descripción general de la HIPAA.](#)

Para obtener información sobre este servicio y la ISO 27001, una norma de administración de la seguridad que especifica las prácticas recomendadas en materia de seguridad, consulte [Información general de ISO 27001](#).

Para obtener información sobre este servicio y el estándar de seguridad de datos del sector de tarjetas de pago (PCI DSS), consulte [Información general de PCI DSS](#).

Para obtener más información sobre este servicio y el estándar del gobierno de Estados Unidos Publicación del Estándar de Procesamiento de la Información Federal (FIPS, Federal Information Processing Standard) 140-2 que especifica los requisitos de seguridad para los módulos criptográficos que protegen información confidencial, consulte la [descripción general del Estándar de Procesamiento de la Información Federal \(FIPS\) 140-2](#) y [Puntos de conexión de Git](#).

Temas

- [Presentamos CodeCommit](#)
- [CodeCommit, Git y elegir el AWS servicio adecuado para sus necesidades](#)
- [¿Cómo CodeCommit funciona?](#)
- [¿En qué se CodeCommit diferencia del control de versiones de archivos en Amazon S3?](#)
- [¿Cómo puedo empezar CodeCommit?](#)

- [¿Dónde puedo encontrar más información sobre Git?](#)

Presentamos CodeCommit

CodeCommit es un servicio de control de código fuente gestionado, seguro y altamente escalable que aloja repositorios Git privados. CodeCommit elimina la necesidad de gestionar tu propio sistema de control de código fuente o de preocuparte por escalar su infraestructura. Se puede utilizar CodeCommit para almacenar cualquier cosa, desde código hasta binarios. Además, es compatible con las funciones estándar de Git, por lo que funciona perfectamente con las herramientas existentes basadas en Git.

Con CodeCommit, puedes:

- Beneficiarse de un servicio totalmente gestionado hospedado por AWS. CodeCommit proporciona una alta disponibilidad y durabilidad del servicio y elimina la sobrecarga administrativa que implica administrar su propio hardware y software. No es necesario aprovisionar ni escalar hardware, ni instalar, configurar y actualizar software de servidor.
- Guarde su código de forma segura. CodeCommit Los repositorios están cifrados tanto en reposo como en tránsito.
- Trabaje en colaboración en el código. CodeCommit Los repositorios admiten solicitudes de incorporación de cambios, que permiten a los usuarios revisar los cambios de código de los demás usuarios y comentarlos antes de combinarlos en sucursales, notificaciones que envían automáticamente correos electrónicos a los usuarios con información sobre las solicitudes de extracción y los comentarios, y mucho más.
- Escale fácilmente sus proyectos de control de versiones. Los repositorios de CodeCommit pueden escalarse para satisfacer sus necesidades de desarrollo. El servicio puede gestionar repositorios con un gran número de archivos o ramificaciones, archivos de gran tamaño e historiales de revisión largos.
- Almacena cualquier cosa en cualquier momento. CodeCommit no tiene límite en cuanto al tamaño de tus repositorios ni a los tipos de archivos que puedes almacenar.
- Intégrelo con otros servicios AWS y con otros servicios de terceros. CodeCommit mantiene sus repositorios cerca de sus otros recursos de producción en la AWS nube, lo que ayuda a aumentar la velocidad y la frecuencia de su ciclo de vida de desarrollo. Está integrado con IAM y se puede utilizar con otros AWS servicios y en paralelo con otros repositorios. Para obtener más información, consulte [Integraciones de productos y servicios con AWS CodeCommit](#).

- Migrar fácilmente archivos desde otros repositorios remotos. Puedes migrar CodeCommit desde cualquier repositorio basado en Git.
- Usa las herramientas de Git que ya conoces. CodeCommit admite los comandos de Git, así como sus propios AWS CLI comandos y APIs.

CodeCommit, Git y elegir el AWS servicio adecuado para sus necesidades

Como servicio basado en Git, CodeCommit se adapta bien a la mayoría de las necesidades de control de versiones. No hay límites arbitrarios en el tamaño de archivo, el tipo de archivo y el tamaño de repositorio. Sin embargo, existen limitaciones inherentes a Git que pueden afectar negativamente al rendimiento de ciertos tipos de operaciones, en concreto a lo largo del tiempo. Puedes evitar la posible degradación del rendimiento del CodeCommit repositorio si evitas usarlo en casos de uso en los que otros AWS servicios se adapten mejor a la tarea. También puede optimizar el rendimiento de Git para repositorios complejos. Estos son algunos casos de uso en los que Git CodeCommit, por lo tanto, podría no ser la mejor solución para ti, o en los que podrías necesitar tomar medidas adicionales para optimizarlo para Git.

caso de uso	Description (Descripción)	Otros servicios que tener en cuenta
Archivos grandes que cambian con frecuencia	Git utiliza codificación delta para almacenar las diferencias entre las versiones de los archivos. Por ejemplo, si cambia unas pocas palabras de un documento, Git solo almacenará las palabras cambiadas. Si tiene archivos u objetos de más de 5 MB con muchos cambios, Git podría necesitar reconstruir una gran cadena de diferencias delta. Esto puede consumir una cantidad cada vez mayor de recursos de cómputo, tanto en	Para versionar archivos de gran tamaño, considere Amazon Simple Storage Service (Amazon S3). Para más información, consulte Uso del control de versiones en la Guía del usuario de Amazon Simple Storage Service.

caso de uso	Description (Descripción)	Otros servicios que tener en cuenta
	el equipo local CodeCommit como en el interno, a medida que estos archivos crecen con el tiempo.	
Base de datos	Los repositorios de Git crecen con el tiempo. Debido a que el control de versiones realiza un seguimiento de todos los cambios, cualquier modificación aumentará el tamaño del repositorio. En otras palabras, a medida que confirma datos, incluso si borra datos en una confirmación, los datos se añaden a un repositorio. A medida que haya más datos para procesar y transmitir con el tiempo, Git se ralentizará. Esto es especialmente perjudicial en un caso de uso de base de datos. Git no se diseñó como una base de datos.	Para crear y utilizar una base de datos con un rendimiento constante, independientemente del tamaño, tenga en cuenta Amazon DynamoDB. Para obtener más información, consulte la Guía de introducción de Amazon DynamoDB.

caso de uso	Description (Descripción)	Otros servicios que tener en cuenta
Registros de seguimiento de auditoría	Normalmente, los registros de seguimiento de auditoría se conservan durante largos períodos de tiempo y se generan continuamente mediante procesos del sistema con una cadencia muy frecuente. Git se diseñó para almacenar de forma segura el código fuente generado por grupos de desarrolladores en un ciclo de desarrollo. Los repositorios que cambian rápidamente y que almacenan continuamente cambios en el sistema generados por programación verán degradado su rendimiento con el tiempo.	Para almacenar los registros de auditoría, considere Amazon Simple Storage Service (Amazon S3). Para auditar AWS la actividad, según tu caso de uso, considera usar AWS CloudTrailAWS Config, o Amazon CloudWatch.

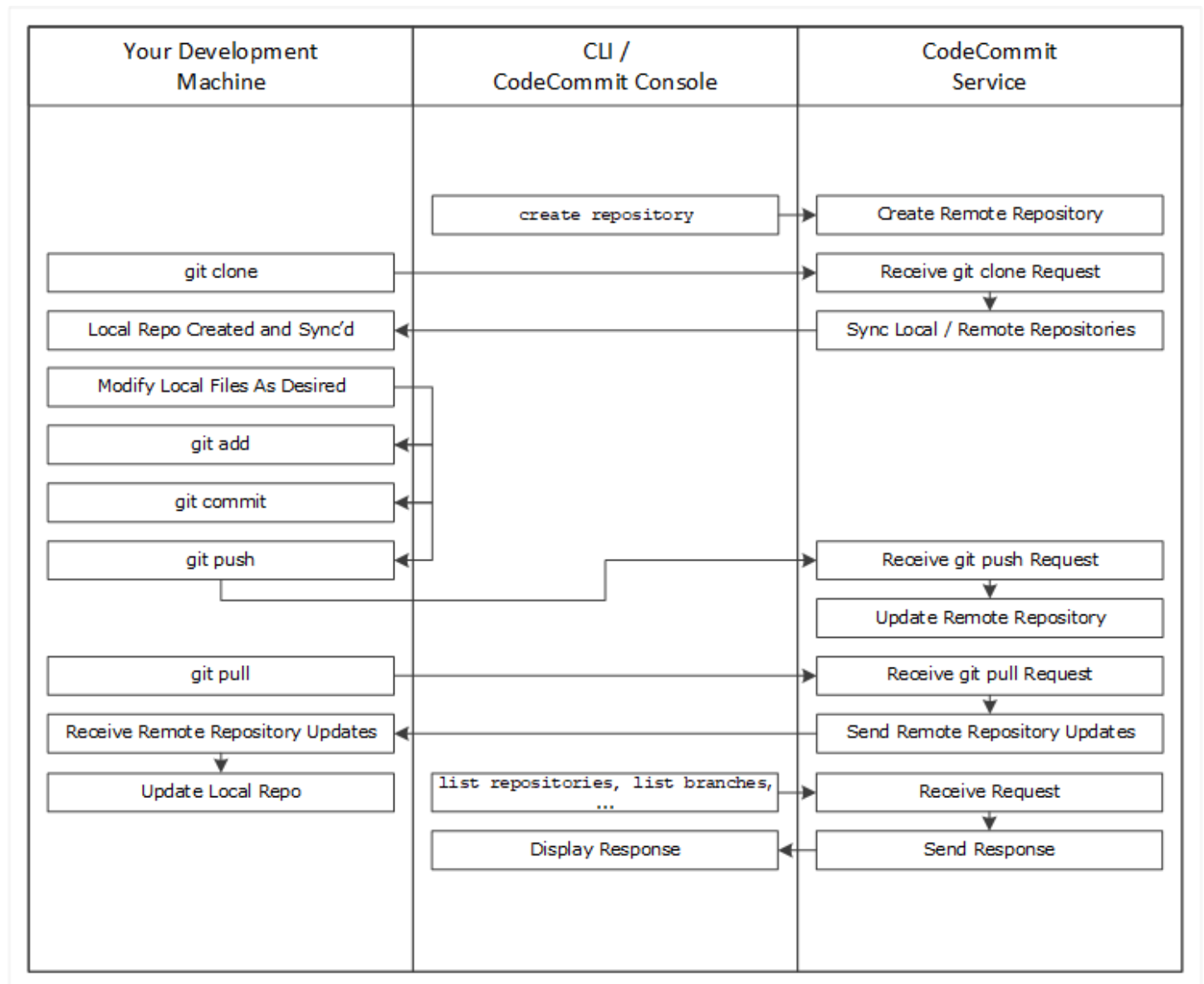
caso de uso	Description (Descripción)	Otros servicios que tener en cuenta
Copias de seguridad	Git se diseñó para el control de versiones del código fuente escrito por desarrolladores. Puedes enviar las confirmaciones a dos repositorios remotos, incluido un CodeCommit repositorio, como estrategia de respaldo. Sin embargo, Git no se diseñó para gestionar las copias de seguridad del sistema de archivos de su equipo, volcados de bases de datos o contenido de copia de seguridad similar. Si se hiciera, podría ralentizar el sistema y aumentar el tiempo necesario para clonar y enviar un repositorio.	Para obtener información sobre cómo realizar copias de seguridad AWS en la nube, consulte Backup & Restore.

caso de uso	Description (Descripción)	Otros servicios que tener en cuenta
Gran cantidad de ramificaciones o referencias	Cuando un cliente Git envía datos al repositorio o extrae datos de él, el servidor remoto debe enviar todas las ramificaciones y referencias, tales como etiquetas, incluso si solo le interesa una sola ramificación. Si tiene miles de ramificaciones y referencias, puede tardar tiempo en el proceso y el envío (negociación de paquetes) y producir una respuesta aparentemente lenta del repositorio. Cuantas más ramificaciones y etiquetas tenga, más tiempo tardará este proceso. Recomendamos usar CodeCommit, pero eliminar, las ramas y etiquetas que ya no sean necesarias.	Para analizar el número de referencias de un CodeCommit repositorio y determinar cuáles podrían no ser necesarias, puedes usar uno de los siguientes comandos: - Emulador de Linux, macOS o Unix o Bash en Windows:<pre>git ls-remote wc -l</pre> - Powershell:<pre>git ls-remote Measure-Object -line</pre>

¿Cómo CodeCommit funciona?

CodeCommit es familiar para los usuarios de repositorios basados en Git, pero incluso aquellos que no estén familiarizados deberían encontrar que la transición a los repositorios CodeCommit es relativamente sencilla. CodeCommit proporciona una consola para crear fácilmente repositorios y enumerar los repositorios y sucursales existentes. En unos pocos pasos sencillos, los usuarios pueden encontrar información sobre un repositorio y clonarla en su ordenador, creando un repositorio local en el que pueden realizar cambios y, a continuación, enviarlos al repositorio. CodeCommit Los usuarios pueden trabajar desde la línea de comandos en sus máquinas locales o utilizar un editor basado en GUI.

La siguiente figura muestra cómo se usa la máquina de desarrollo, la CodeCommit consola AWS CLI o el CodeCommit servicio para crear y administrar repositorios:



1. Utilice la consola AWS CLI o la CodeCommit consola para crear un CodeCommit repositorio.
2. Desde tu máquina de desarrollo, usa Git para ejecutar `git clone`, especificando el nombre del CodeCommit repositorio. Esto crea un repositorio local que se conecta al CodeCommit repositorio.
3. Utilice el repositorio local en su equipo de desarrollo para modificar (añadir, editar y eliminar) archivos y, a continuación, ejecute `git add` para almacenar los archivos modificados localmente. Ejecute `git commit` para confirmar los archivos localmente y, a continuación, ejecute `git push` para enviar los archivos al CodeCommit repositorio.

4. Descargue los cambios de otros usuarios. Ejecute `git pull` para sincronizar los archivos del CodeCommit repositorio con su repositorio local. De este modo, se asegura de que trabaja con la versión más reciente de los archivos.

Puedes usar la consola AWS CLI o la CodeCommit consola para rastrear y administrar tus repositorios.

¿En qué se CodeCommit diferencia del control de versiones de archivos en Amazon S3?

CodeCommit está optimizado para el desarrollo de software en equipo. Gestiona lotes de cambios en varios archivos, cosa que puede realizarse en paralelo con los cambios realizados por otros desarrolladores. El control de versiones de Amazon S3; admite la recuperación de versiones pasadas de archivos, pero no se centra en las características de seguimiento de archivos colaborativas que necesitan los equipos de desarrollo de software.

¿Cómo puedo empezar CodeCommit?

Para empezar con CodeCommit:

1. Siga los pasos que se detallan en [Configuración](#) para preparar los equipos de desarrollo.
2. Siga los pasos que se detallan en uno o más de los tutoriales de [Introducción](#).
3. [Cree](#) proyectos de control de versiones CodeCommit o [migre](#) proyectos de control de versiones a CodeCommit.

¿Dónde puedo encontrar más información sobre Git?

Si no conoce Git todavía, debe [aprender a utilizar Git](#). A continuación encontrará algunos recursos útiles:

- [Pro Git](#) es una versión online del libro Pro Git escrito por Scott Chacon y publicado por Apress.
- [Git Immersion](#), una visita try-it-yourself guiada que te explica los aspectos básicos del uso de Git. publicada por Neo Innovation, Inc.
- [Git Reference](#) es una referencia rápida online que también puede usarse como tutorial en profundidad sobre Git Publicado por el GitHub equipo.

- [Git Cheat Sheet](#) incluye la sintaxis de comandos de Git básicos Publicado por el GitHub equipo.
- [Git Pocket Guide](#) escrita por Richard E. Silverman. Publicada por O'Reilly Media, Inc.

Configuración de AWS CodeCommit

Puede iniciar sesión en la Consola de administración de AWS y [cargar, añadir o editar un archivo](#) en un repositorio directamente desde la consola de AWS CodeCommit. Se trata de una forma rápida de realizar un cambio. Sin embargo, si desea trabajar con varios archivos, archivos en todas las ramificaciones, etc., considere la posibilidad de configurar su equipo local para trabajar con repositorios. La forma más sencilla de configurar CodeCommit consiste en definir las credenciales de Git HTTPS para AWS CodeCommit. Este método de autenticación HTTPS:

- Utiliza un nombre de usuario y contraseña estáticas
- Funciona con todos los sistemas operativos compatibles con CodeCommit.
- También es compatible con entornos de desarrollo integrado (IDE) y otras herramientas de desarrollo que admiten las credenciales de Git

Puede utilizar otros métodos si no desea o no puede utilizar credenciales de Git por razones operativas. Por ejemplo, si accede a repositorios de CodeCommit mediante acceso federado, credenciales temporales o un proveedor de identidad web, no puede usar credenciales de Git. Le recomendamos que configure el equipo local con el comando `git-remote-codecommit`. Examine detenidamente las siguientes opciones para decidir qué método alternativo se adapta mejor a sus necesidades.

- [Configuración mediante credenciales de Git](#)
- [Configuración con otros métodos](#)
- [Compatibilidad entre CodeCommit, Git y otros componentes](#)

Para obtener información sobre cómo utilizar CodeCommit y Amazon Virtual Private Cloud, consulte [Utilización AWS CodeCommit con puntos finales de VPC de interfaz](#).

Visualización y administración de sus credenciales

Puede ver y administrar sus credenciales de CodeCommit desde la consola AWS a través de Mis credenciales de seguridad.

 Note

Esta opción no está disponible para los usuarios que utilizan acceso federado, credenciales temporales o un proveedor de identidad web.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En la esquina superior derecha de la barra de navegación, elija su nombre de usuario y, a continuación, elija Security Credentials (Credenciales de seguridad).
3. Seleccione la pestaña de credenciales de AWS CodeCommit.

Configuración mediante credenciales de Git

Con conexiones HTTPS y credenciales de Git, puede generar un nombre de usuario y contraseña estáticos en IAM. Seguidamente, puede utilizar estas credenciales con Git y cualquier herramienta de terceros que admita la autenticación con nombre de usuario y contraseña de Git. Este método es compatible con la mayoría de IDE y herramientas de desarrollo. Es el método de conexión más fácil y sencillo para CodeCommit.

- [Para usuarios HTTPS mediante credenciales de Git](#): siga estas instrucciones para establecer conexiones entre su equipo local y los repositorios de CodeCommit utilizando credenciales de Git.
- [Para conexiones desde herramientas de desarrollo](#): siga estas instrucciones para establecer conexiones entre su IDE u otras herramientas de desarrollo y los repositorios de CodeCommit utilizando credenciales de Git. Los IDE que admiten las credenciales de Git son, entre otros, Visual Studio, Xcode e IntelliJ.

Configuración con otros métodos

Puede usar el protocolo SSH en lugar de HTTPS para conectarse a su repositorio de CodeCommit. Con conexiones SSH, debe crear archivos de claves públicas y privadas en su equipo local que Git y CodeCommit utilizarán para la autenticación SSH. Puede asociar la clave pública con su usuario de IAM. Puede almacenar la clave privada en su equipo local. Dado que SSH requiere la creación y gestión manual de archivos de claves públicas y privadas, es posible que las credenciales de Git sean más fáciles y sencillas de utilizar con CodeCommit.

A diferencia de las credenciales de Git, la configuración de la conexión SSH varía en función del sistema operativo de su equipo local.

- [Para los usuarios de SSH que no utilizan AWS CLI](#): siga estas instrucciones abreviadas si ya dispone de un par de claves pública y privada y conoce las conexiones SSH de su equipo local.
- [Para conexiones SSH en Linux, macOS o Unix](#): siga estas instrucciones para obtener una explicación detallada sobre el proceso de creación de un par de claves pública y privada y sobre la configuración de conexiones en sistemas operativos Linux, macOS o Unix.
- [Para conexiones SSH en Windows](#): siga estas instrucciones para obtener una explicación detallada sobre el proceso de creación de un par de claves pública y privada y sobre la configuración de conexiones en sistemas operativos Windows.

Si se conecta a CodeCommit y AWS mediante el acceso federado, un proveedor de identidad o credenciales temporales, o si no desea configurar usuarios de IAM o credenciales de Git para usuarios de IAM, puede configurar las conexiones a los repositorios de CodeCommit de una de estas dos maneras:

- Instalar y usar `git-remote-codecommit` (recomendado).
- Instalar y utilizar la aplicación auxiliar de credenciales incluida en la AWS CLI.

Ambos métodos admiten el acceso a los repositorios de CodeCommit sin necesidad de un usuario de IAM, lo que significa que puede conectarse a los repositorios mediante acceso federado y credenciales temporales. La utilidad `git-remote-codecommit` es el enfoque recomendado. Amplía la funcionalidad de Git y es compatible con varias versiones de Git y aplicaciones auxiliares de credenciales. Sin embargo, no todos los IDE admiten el formato de URL de clonación utilizado por `git-remote-codecommit`. Es posible que tenga que clonar manualmente los repositorios en el equipo local antes de poder trabajar con ellos en el IDE.

- Siga las instrucciones de [Pasos de configuración para conexiones HTTPS a repositorios AWS CodeCommit con git-remote-codecommit](#) para instalar y configurar `git-remote-codecommit` en Windows, Linux, macOS o Unix.

La aplicación de ayudante de credenciales incluida en la AWS CLI permite que Git use HTTPS y una versión con firma criptográfica de sus credenciales de usuario de IAM o del rol de la instancia de Amazon EC2 cada vez que Git requiera autenticarse con AWS para interactuar con los repositorios de CodeCommit. Algunos sistemas operativos y versiones de Git disponen de su propia aplicación

auxiliar de credenciales, que es incompatible con la aplicación auxiliar incluida en la AWS CLI. Pueden provocar problemas de conectividad en CodeCommit.

- [Para conexiones HTTPS en Linux, macOS o Unix con el asistente de AWS CLI credenciales](#): siga estas instrucciones para obtener una explicación detallada del proceso de instalación y configuración de la aplicación ayudante de credenciales en sistemas Linux, macOS o Unix.
- [Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI](#) : siga estas instrucciones para obtener una explicación detallada del proceso de instalación y configuración del auxiliar de credenciales en sistemas Windows.

Si se conecta a un repositorio de CodeCommit que se aloja en otra cuenta de Amazon Web Services, puede configurar el acceso y las conexiones mediante funciones, políticas y la aplicación ayudante de credenciales incluida en la AWS CLI.

- [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#): siga estas instrucciones para obtener una explicación detallada del proceso de configuración del acceso entre cuentas en una cuenta de Amazon Web Services a los usuarios de un grupo de IAM en otra cuenta de Amazon Web Services.

Compatibilidad entre CodeCommit, Git y otros componentes

Al trabajar con CodeCommit, utiliza Git. Y es posible que utilice otros programas. La siguiente tabla incluye la lista actualizada de compatibilidades de versiones. Como práctica recomendada, le recomendamos que use las últimas versiones de Git y otro software.

Información de compatibilidad de versiones de AWS CodeCommit

Componente	Versión
Git	CodeCommit admite las versiones de Git 1.7.9 y posteriores. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.
Curl	CodeCommit requiere curl 7.33 y versiones posteriores. Sin embargo, existe un problema conocido con HTTPS y la actualización

Componente	Versión
	7.41.0 de curl. Para obtener más información, consulte Solución de problemas .
Python (solo git-remote-codecommit)	git-remote-codecommit requiere la versión 3 y posterior.
Pip (solo git-remote-codecommit)	git-remote-codecommit requiere la versión 9.0.3 y posterior.
AWS CLI (git-remote-codecommit solamente)	Recomendamos una versión reciente de AWS CLI versión 2 para todos los usuarios de CodeCommit. git-remote-codecommit requiere AWS CLI versión 2 para admitir el SSO de AWS y las conexiones que requieren credenciales temporales, como los usuarios federados.

Configuración de usuarios HTTPS mediante credenciales de Git

La forma más sencilla de configurar las conexiones a AWS CodeCommit los repositorios es configurar las credenciales de Git CodeCommit en la consola de IAM y, a continuación, utilizarlas para las conexiones HTTPS. También puede utilizar estas credenciales en cualquier herramienta de terceros o entorno de desarrollo integrado (IDE) compatible con la autenticación HTTPS mediante un nombre de usuario y contraseña estáticos. Para ver ejemplos, consulte [Para conexiones desde herramientas de desarrollo](#).

Note

Si has configurado previamente tu ordenador local para usar el asistente de credenciales CodeCommit, debes editar tu archivo.gitconfig para eliminar la información del asistente de credenciales del archivo antes de poder usar las credenciales de Git. Si el equipo local ejecuta macOS, es posible que deba borrar las credenciales de Keychain Access almacenadas en caché.

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para configurar una cuenta de Amazon Web Services, crear un usuario de IAM y configurar el acceso a CodeCommit.

Para crear y configurar un usuario de IAM al que acceder CodeCommit

1. Cree una cuenta de Amazon Web Services yendo a <http://aws.amazon.com> y seleccionando Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUser o otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Si desea utilizar AWS CLI comandos con CodeCommit, instale el AWS CLI. Le recomendamos que cree un perfil para usarlo AWS CLI con CodeCommit. Para obtener más información, consulte [Referencia de la línea de comandos](#) en [Uso de perfiles con nombre](#).

Paso 2: Instalar Git

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Note

Git es una plataforma en evolución que se actualiza periódicamente. En ocasiones, un cambio en una función puede afectar a su funcionamiento. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Paso 3: Crear credenciales de Git para las conexiones HTTPS a CodeCommit

Después de instalar Git, cree credenciales de Git para el usuario de IAM en IAM.

Para configurar las credenciales de Git HTTPS para CodeCommit

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.

Asegúrese de iniciar sesión como el usuario de IAM que creará y usará las credenciales de Git para CodeCommit las conexiones.

2. En el panel de navegación de la consola de IAM, seleccione Usuarios y, en la lista de usuarios seleccione su usuario de IAM.

Note

Puedes ver y gestionar tus CodeCommit credenciales directamente en Mis credenciales de seguridad. Para obtener más información, consulte [Visualización y administración de sus credenciales](#).

3. En la página de detalles del usuario, seleccione la pestaña Credenciales de seguridad. En la sección Claves de API, selecciona Generar clave de API. Selecciona AWS CodeCommit como servicio.

▼ Access Management

Roles

Policies

IAM users

IAM user groups

Identity providers

Account settings

Root access management

Temporary delegation requests

▼ Access reports

Access Analyzer

API keys (0)

Actions ▼

Generate API Key

API keys grant access to specific AWS services. You can have a maximum of two API keys per service (active, inactive, or expired) at a time. [Learn more](#)

Service
Any ▼

Status
Any ▼

Service ▼	API key name ▼	Created ▼	Expires ▼	Status ▼

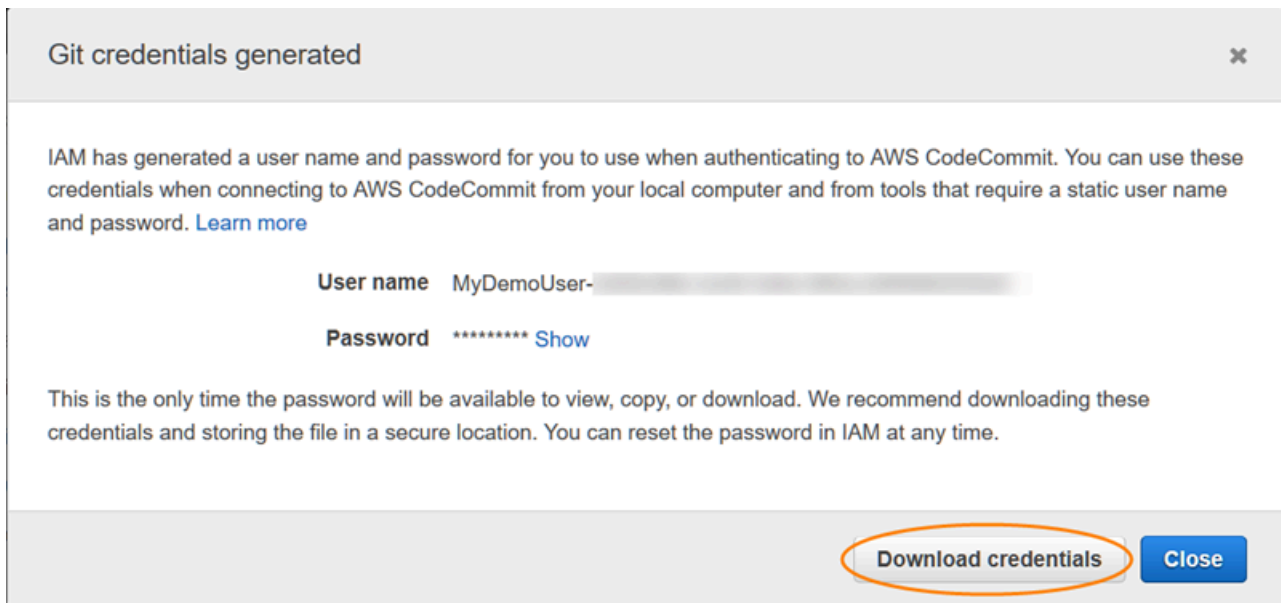
No API keys

Generate API Key

Note

No puede elegir su propio nombre de usuario o contraseña para las credenciales de Git. Para obtener más información, consulta [Usar credenciales de Git y HTTPS con CodeCommit](#).

4. Copie el nombre de usuario y la contraseña que IAM ha generado para usted (puede mostrar, copiar y, a continuación, pegar la información en un archivo seguro en el equipo local), o seleccione Descargar credenciales para descargar la información en un archivo .CSV. Necesitará esta información para conectarse a CodeCommit.



Una vez que haya guardado las credenciales, elija Cerrar.

 Important

Esta es la única oportunidad de guardar el nombre de usuario y la contraseña. Si no lo hace, podrá copiar el nombre de usuario de la consola de IAM, pero no podrá buscar la contraseña. Deberá restablecer la contraseña y, a continuación, guardarla.

Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya le ha enviado el nombre y los detalles de conexión del repositorio de CodeCommit, puede omitir este paso y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.

- Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
- Copie la dirección URL HTTPS (GRC) si utiliza el comando `git-remote-codecommit` en su equipo local.
- Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

 Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. Abra un terminal, una línea de comandos o un shell de Git. Ejecute el comando `git clone` con la URL de clonación HTTPS que copió para clonar el repositorio. Por ejemplo, para clonar un repositorio *MyDemoRepo* con el nombre de un repositorio local denominado *my-demo-repo* en la región EE.UU. Este (Ohio):

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

La primera vez que se conecte, se le pedirá el nombre de usuario y la contraseña para el repositorio. Según la configuración del equipo local, este mensaje se envía desde el sistema de gestión de credenciales del sistema operativo, la utilidad de administración de credenciales de la versión de Git (por ejemplo, el Administrador de credenciales de Git de Windows), el IDE o Git. Introduzca el nombre de usuario y la contraseña de las credenciales de Git que generó en IAM (las que creó en [Paso 3: Crear credenciales de Git para las conexiones HTTPS a CodeCommit](#)). En función del sistema operativo y otro software, esta información puede guardarse en un almacén de credenciales o en la utilidad de Administración de credenciales. Si se guarda, esta información no se le volverá a pedir a menos que cambie la contraseña, inactive las credenciales de Git o elimine las credenciales de Git en IAM.

Si el equipo local no tiene configurado un almacén de credenciales o una utilidad de Administración de credenciales, puede instalar uno. Para obtener más información sobre Git y

cómo gestiona las credenciales, consulte [Almacenamiento de credenciales](#) en la documentación de Git.

Para obtener más información, consulte [Conéctese al CodeCommit repositorio clonando el repositorio](#) y [Creación de una confirmación](#).

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Para aprender a crear e insertar su primera confirmación, consulte [Crea una confirmación en AWS CodeCommit](#). Si no tiene experiencia en Git, es posible que quiera consultar la información de [¿Dónde puedo encontrar más información sobre Git?](#) y [Primeros pasos con Git y AWS CodeCommit](#).

Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit

Si desea conectarse CodeCommit mediante una cuenta raíz, un acceso federado o credenciales temporales, debe configurar el acceso mediante. git-remote-codecommit Esta utilidad proporciona un método sencillo para insertar y extraer código de los CodeCommit repositorios mediante la extensión de Git. Es el método recomendado para admitir conexiones realizadas con acceso federado, proveedores de identidad y credenciales temporales. Para asignar permisos a identidades federadas, debe crear un rol y definir permisos para este. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte [Crear un rol para un proveedor de identidad de terceros \(federación\)](#) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

También se puede utilizar git-remote-codecommit con un usuario de IAM. A diferencia de otros métodos de conexión HTTPS, git-remote-codecommitno requiere la configuración de credenciales de Git para el usuario.

 Note

Algunos IDEs no admiten el formato de URL de clonación utilizado por `git-remote-codecommit`. Es posible que tenga que clonar manualmente los repositorios en el equipo local para poder trabajar con ellos en el IDE de su elección. Para obtener más información, consulte [Solución de problemas git-remote-codecommit y AWS CodeCommit](#).

Estos procedimientos se redactan partiendo del supuesto de que dispone de una cuenta de Amazon Web Services, ha creado al menos un repositorio en él CodeCommit y utiliza un usuario de IAM con una política gestionada al conectarse a los CodeCommit repositorios. Para obtener información acerca de cómo configurar el acceso para usuarios federados y otros tipos de credenciales rotativas, consulte [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#).

Temas

- [Paso 0: Instalar los requisitos previos para git-remote-codecommit](#)
- [Paso 1: Configuración inicial para CodeCommit](#)
- [Paso 2: Instalar git-remote-codecommit](#)
- [Paso 3: Conectarse a la CodeCommit consola y clonar el repositorio](#)
- [Sigüientes pasos](#)

Paso 0: Instalar los requisitos previos para git-remote-codecommit

Para poder utilizar `git-remote-codecommit`, debe instalar algunos componentes necesarios en el equipo local. Entre ellos se incluyen:

- Una versión compatible de Python. Para obtener más información sobre las versiones compatibles de Python, consulte [git-remote-codecommit](#).

Para obtener más información sobre la configuración y el uso `git-remote-codecommit`, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

- Git

 Note

Cuando instale Python en Windows, asegúrese de elegir la opción para agregar Python a la ruta.

git-remote-codecommit requiere pip versión 9.0.3 o posterior. Para comprobar su versión de pip, abra un terminal o línea de comandos y ejecute el siguiente comando:

```
pip --version
```

Puede ejecutar los dos comandos siguientes para actualizar su versión de pip a la última versión:

```
curl -O https://bootstrap.pypa.io/get-pip.py  
python3 get-pip.py --user
```

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. En ocasiones, un cambio en una función puede afectar a su funcionamiento. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para crear un usuario de IAM, configurarlo con las políticas correspondientes, obtener una clave de acceso y una clave secreta e instalar y configurar la AWS CLI.

Para crear y configurar un usuario de IAM para acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Para instalar y configurar el AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con él CodeCommit desde la línea de comandos. Le recomendamos que instale la versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales congit-remote-codecommit.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplous-east-2. Cuando se le pregunte el formato de salida predeterminado, indique json. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
```

Default region name [None]: *Type a supported region for CodeCommit here, and then press Enter*

Default output format [None]: *Type json here, and then press Enter*

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)
- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- sa-east-1
- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1

- us-gov-west-1
- us-gov-east-1
- eu-north-1
- ap-east-1
- me-south-1
- cn-north-1
- cn-northwest-1
- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

Paso 2: Instalar git-remote-codecommit

Siga estos pasos para instalar git-remote-codecommit.

Para instalar git-remote-codecommit

1. En el terminal o en la línea de comandos, ejecute el comando:

```
pip install git-remote-codecommit
```

Note

Según el sistema operativo y la configuración, es posible que deba ejecutar este comando con permisos elevados, como sudo, o usar el parámetro --user para realizar la instalación en un directorio que no requiera privilegios especiales, como su cuenta de usuario actual. Por ejemplo, en una computadora con Linux, macOS o Unix:

```
sudo pip install git-remote-codecommit
```

En un equipo con Windows:

```
pip install --user git-remote-codecommit
```

2. Supervise el proceso de instalación hasta que vea un mensaje de éxito.

Paso 3: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya te ha enviado la URL del clon git-remote-codecommit para usarla en el CodeCommit repositorio, puedes omitir la conexión a la consola y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.
 - Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
 - Copie la dirección URL HTTPS (GRC) si utiliza el comando git-remote-codecommit en su equipo local.
 - Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas

iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. En el terminal o símbolo del sistema, clone el repositorio con el comando `git clone`. Usa la `git-remote-codecommit` URL HTTPS que copiaste y el nombre del AWS CLI perfil, si has creado un perfil con nombre. Si no especifica un perfil, el comando presupone que el perfil es el predeterminado. El repositorio local se crea en un subdirectorio del directorio en el que ejecuta el comando. Por ejemplo, para clonar un repositorio denominado *MyDemoRepo* en un repositorio local llamado *my-demo-repo*:

```
git clone codecommit://MyDemoRepo my-demo-repo
```

Para clonar el mismo repositorio mediante un perfil denominado *CodeCommitProfile*:

```
git clone codecommit://CodeCommitProfile@MyDemoRepo my-demo-repo
```

Para clonar un repositorio en un repositorio Región de AWS diferente al configurado en su perfil, incluya el Región de AWS nombre. Por ejemplo:

```
git clone codecommit::ap-northeast-1://MyDemoRepo my-demo-repo
```

Siguientes pasos

Ha completado los requisitos previos. Siga los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Para aprender a crear e insertar su primera confirmación, consulte [Crea una confirmación en AWS CodeCommit](#). Si no tiene experiencia en Git, es posible que quiera consultar la información de [¿Dónde puedo encontrar más información sobre Git?](#) y [Primeros pasos con Git y AWS CodeCommit](#).

Configuración de conexiones desde herramientas de desarrollo mediante credenciales de Git

Una vez que haya configurado las credenciales de Git de AWS CodeCommit en la consola de IAM, puede utilizar dichas credenciales en cualquier herramienta de desarrollo compatible con las credenciales de Git. Por ejemplo, puede configurar el acceso al repositorio de CodeCommit en AWS

Cloud9, Visual Studio, Xcode, IntelliJ o cualquier entorno de desarrollo integrado (IDE) que integre las credenciales de Git. Una vez que haya configurado el acceso, puede editar el código, confirmar los cambios y enviarlos directamente desde el IDE o cualquier otra herramienta de desarrollo.

Note

Si accede a los repositorios de CodeCommit mediante acceso federado, credenciales temporales o un proveedor de identidad web, no puede usar las credenciales de Git. Le recomendamos que configure el equipo local con el comando `git-remote-codecommit`. Sin embargo, no todos los IDE son totalmente compatibles con el código auxiliar remoto de Git como `git-remote-codecommit`. Si surgen problemas, consulte [Solución de problemas git-remote-codecommit y AWS CodeCommit](#).

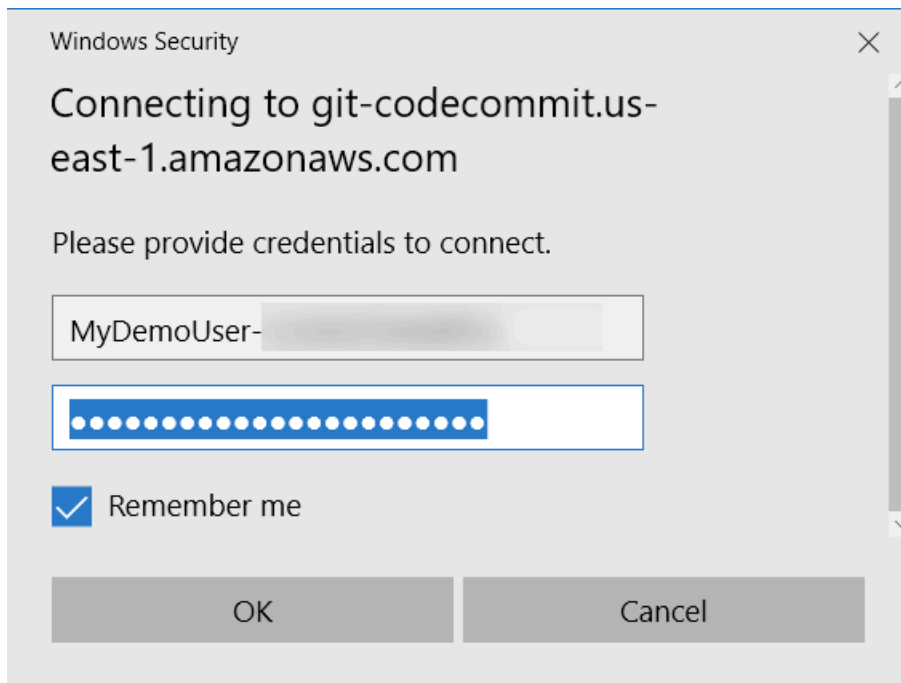
Temas

- [Integre AWS Cloud9 con AWS CodeCommit](#)
- [Integrar Visual Studio con AWS CodeCommit](#)

Cuando el IDE o la herramienta de desarrollo le pida el nombre de usuario y la contraseña utilizados para conectarse al repositorio de CodeCommit, introduzca las credenciales de Git para Nombre de usuario y Contraseña que creó en IAM.

Para obtener más información sobre las Regiones de AWS y puntos de conexión de CodeCommit, consulte [Regiones y puntos de conexión de Git](#).

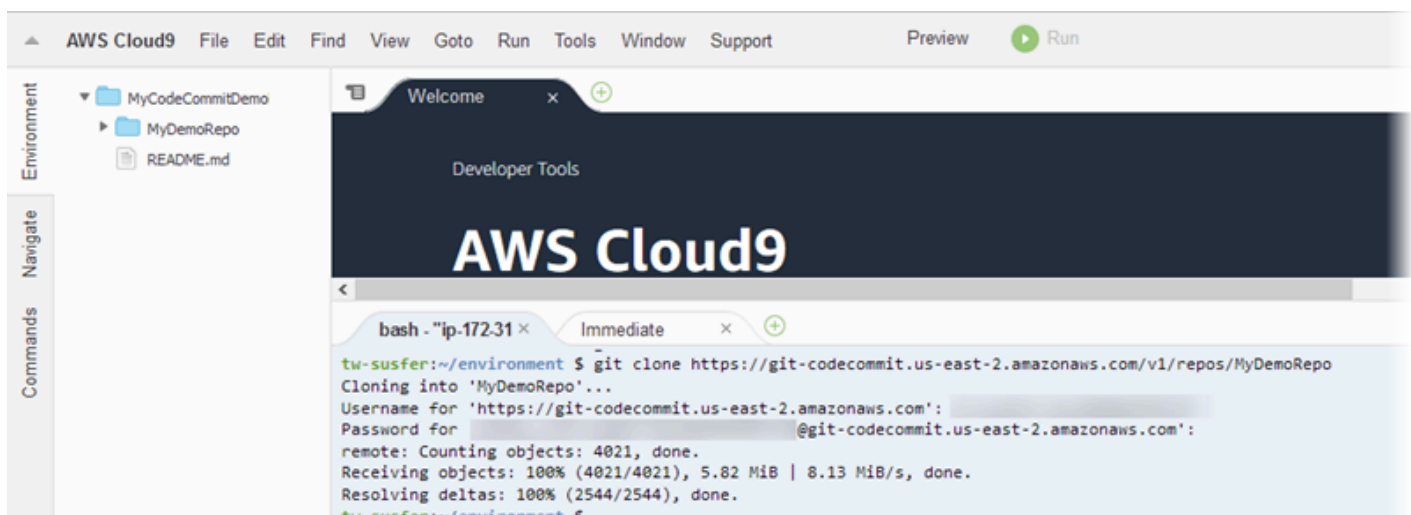
También se le puede pedir que almacene el nombre de usuario y la contraseña desde el sistema operativo. Por ejemplo, en Windows, debería introducir las credenciales de Git de la siguiente forma:



Para obtener información sobre la configuración de las credenciales de Git para un determinado programa o herramienta de desarrollo de software, consulte la documentación del producto.

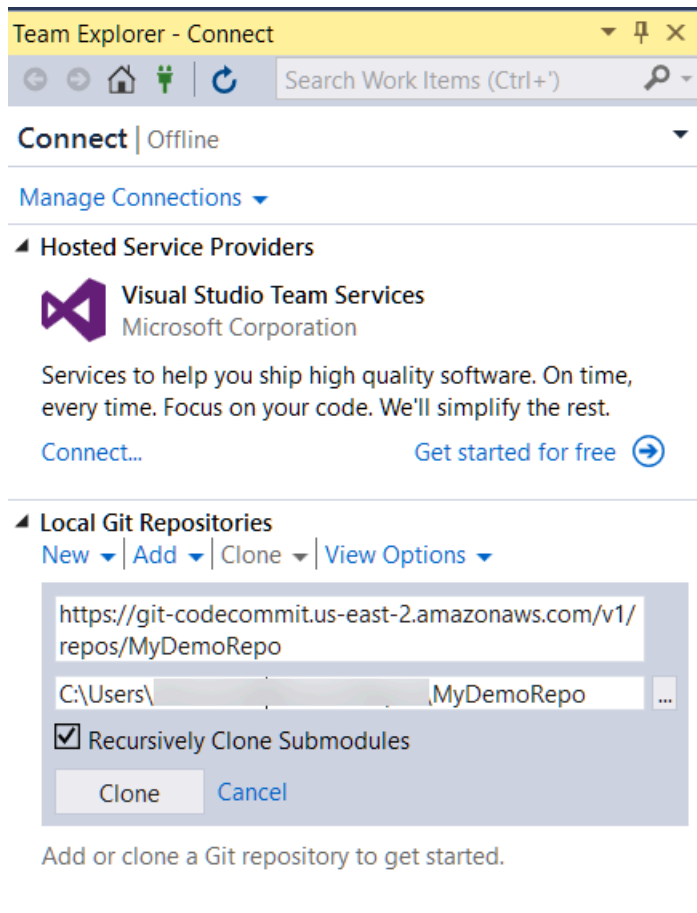
A continuación, no se ofrece una lista completa de IDE. Los enlaces se proporcionan únicamente para ayudarle a obtener más información sobre estas herramientas. AWS no se hace responsable del contenido de ninguno de estos temas.

- [AWS Cloud9](#)



- [Visual Studio](#)

De forma alternativa, instale el AWS Toolkit for Visual Studio. Para obtener más información, consulte [Integrar Visual Studio con AWS CodeCommit](#).



- [XCode](#)

Integre AWS Cloud9 con AWS CodeCommit

Se puede utilizar AWS Cloud9 para realizar cambios de código en un CodeCommit repositorio. AWS Cloud9 contiene un conjunto de herramientas que puede utilizar para escribir código y compilar, ejecutar, probar, depurar y publicar software. Puede clonar repositorios existentes, crear repositorios, confirmar y enviar cambios de código a un repositorio, etc., todo ello desde su entorno de desarrollo de AWS Cloud9 EC2. El entorno de desarrollo de AWS Cloud9 EC2 suele estar preconfigurado con un rol AWS CLI de Amazon EC2 y Git, por lo que, en la mayoría de los casos, puede ejecutar algunos comandos sencillos y empezar a interactuar con su repositorio.

Para usarlo AWS Cloud9 con CodeCommit, necesita lo siguiente:

- Un entorno de desarrollo AWS Cloud9 EC2 que se ejecuta en Amazon Linux.

- El AWS Cloud9 IDE se abre en un navegador web.
- Un usuario de IAM al que se le ha aplicado una de las políticas CodeCommit gestionadas y una de las políticas AWS Cloud9 gestionadas.

Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#) y [Descripción y obtención de las credenciales de seguridad](#).

Note

En este tema se describe la configuración de la integración con CodeCommit y AWS Cloud9 con el acceso general desde Internet. Puede configurar el acceso a CodeCommit y AWS Cloud9 en un entorno aislado, pero eso requiere pasos adicionales. Para obtener más información, consulte lo siguiente:

- [Utilización AWS CodeCommit con puntos finales de VPC de interfaz](#)
- [Acceso a instancias Amazon EC2 sin ingreso con AWS Systems Manager](#)
- [Trabajo con entornos compartidos](#)
- [Compartir la VPC con otras cuentas](#)
- [Entrada de blog: Aislar el acceso a la red a sus entornos AWS Cloud9](#)

Temas

- [Paso 1: Crear un entorno de AWS Cloud9 desarrollo](#)
- [Paso 2: Configure el asistente de AWS CLI credenciales en su AWS Cloud9 entorno de desarrollo de EC2](#)
- [Paso 3: Clone un CodeCommit repositorio en su entorno de desarrollo de EC2 AWS Cloud9](#)
- [Siguiendo pasos](#)

Paso 1: Crear un entorno de AWS Cloud9 desarrollo

AWS Cloud9 aloja su entorno de desarrollo en una instancia de Amazon EC2. Esta es la forma más sencilla de integración, ya que puede usar las credenciales temporales AWS administradas de la instancia para conectarse a su CodeCommit repositorio. Si desea utilizar su propio servidor en su lugar, consulte la [Guía de usuario de AWS Cloud9](#).

Para crear un AWS Cloud9 entorno

1. Inicie sesión AWS como el usuario de IAM que ha configurado y abra la AWS Cloud9 consola.
2. En la AWS Cloud9 consola, selecciona Crear entorno.
3. En Step 1: Name environment (Paso 1: Asignar nombre al entorno), introduzca un nombre y una descripción opcional para el entorno y, a continuación, elija Next step (Paso siguiente).
4. En Step 2: Configure Settings, configure el entorno como se indica a continuación:
 - En Environment type, elija Create a new instance for environment (EC2).
 - En Instance type, elija el tipo de instancia adecuado para su entorno de desarrollo. Por ejemplo, si solo está explorando el servicio, puede elegir la opción predeterminada, t2.micro. Si va a utilizar este entorno para trabajos de desarrollo, elija un tipo de instancia más grande.
 - Acepte las demás configuraciones predeterminadas a menos que tenga motivos para elegir otra opción (por ejemplo, su organización utiliza una VPC específica o su cuenta de Amazon Web Services no tiene ninguna VPCs configurada) y, a continuación, seleccione Siguiente paso.
5. En Step 3: Review, revise la configuración. Si necesita realizar cambios, elija Previous step. En caso contrario, elija Create environment.

Se tardan varios minutos en crear un entorno y conectarse a él por primera vez. Si le parece que tarda demasiado, consulte [Solución de problemas](#) en la Guía del usuario de AWS Cloud9 .

6. Cuando esté conectado a su entorno, ejecute el comando `git --version` en la ventana de terminal para comprobar si Git ya está instalado y es una versión compatible.

Si Git no está instalado, o si no es una versión compatible, instala una versión compatible. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git. Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Tip

En función del sistema operativo de su entorno, es posible que pueda utilizar el comando `yum` con la opción `sudo` para instalar actualizaciones, incluido Git. Por ejemplo, una secuencia de comandos administrativos podría ser como los tres comandos siguientes:

```
sudo yum -y update
```



```
sudo yum -y install git
git --version
```

7. Configure el nombre de usuario y el correo electrónico que se van a asociar a sus comandos Git ejecutando el comando `git config`. Por ejemplo:

```
git config --global user.name "Mary Major"
git config --global user.email mary.major@example.com
```

Paso 2: Configure el asistente de AWS CLI credenciales en su AWS Cloud9 entorno de desarrollo de EC2

Después de crear un AWS Cloud9 entorno, puede configurar el asistente de AWS CLI credenciales para administrar las credenciales de las conexiones a su repositorio. CodeCommit El entorno de AWS Cloud9 desarrollo incluye credenciales temporales AWS gestionadas que están asociadas a su usuario de IAM. Estas credenciales se utilizan con el asistente de AWS CLI credenciales.

1. Abra la ventana de terminal y ejecute el siguiente comando para comprobar que la AWS CLI está instalada:

```
aws --version
```

Si se ejecuta correctamente, este comando devuelve la versión de la AWS CLI instalada actualmente. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación](#) del. AWS Command Line Interface

2. En la terminal, ejecuta los siguientes comandos para configurar el asistente de AWS CLI credenciales para las conexiones HTTPS:

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

Tip

El ayudante de credenciales utiliza el rol predeterminado de la instancia de Amazon EC2 para el entorno de desarrollo. Si piensa utilizar el entorno de desarrollo para conectarse

a repositorios que no están alojados CodeCommit, configure las conexiones SSH a esos repositorios o configure un `.gitconfig` archivo local para utilizar un sistema de administración de credenciales alternativo al conectarse a esos otros repositorios. Para obtener más información, consulte [Git Tools - Credential Storage](#) en el sitio web de Git.

Paso 3: Clone un CodeCommit repositorio en su entorno de desarrollo de EC2 AWS Cloud9

Una vez que haya configurado el asistente de AWS CLI credenciales, puede clonar su CodeCommit repositorio en él. A continuación, puede comenzar a trabajar con el código.

1. En el terminal, ejecute el comando `git clone`, especificando la URL HTTPS de clonación del repositorio que desea clonar. Por ejemplo, si quieres clonar un repositorio cuyo nombre pertenece a la MyDemoRepo región Este de EE. UU. (Ohio), debes introducir:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Tip

Para encontrar la URL de clonación de tu repositorio en la CodeCommit consola, selecciona la URL de clonación.

2. Cuando haya finalizado la clonación, expanda la carpeta del repositorio en el panel de navegación lateral y elija el archivo que desea abrir para editarlo. También puede elegir File (Archivo) seguido de New File (Nuevo archivo) para crear un archivo.
3. Cuando haya terminado de editar o crear archivos, en la ventana de terminal, desplácese al directorio que contiene el repositorio clonado y, a continuación, confirme y envíe los cambios. Por ejemplo, si has añadido un archivo nuevo llamado *MyFile.py*:

```
cd MyDemoRepo
git commit -a MyFile.py
git commit -m "Added a new file with some code improvements"
git push
```

Siguientes pasos

Para obtener más información, consulte la [Guía del AWS Cloud9 usuario](#) y el [CodeCommit ejemplo de AWS Cloud9](#). Para obtener más información sobre el uso de Git con CodeCommit, consulte [Primeros pasos con Git y AWS CodeCommit](#).

Integrar Visual Studio con AWS CodeCommit

Puede utilizar Visual Studio para realizar cambios en el código de un repositorio de CodeCommit. AWS Toolkit for Visual Studio ahora incluye características que hacen más fácil y cómodo el uso de CodeCommit cuando se trabaja en Visual Studio. La integración con Toolkit for Visual Studio está diseñada para funcionar con credenciales de Git y un usuario de IAM. Puede clonar repositorios existentes, crear repositorios, confirmar y enviar cambios en el código a un repositorio y mucho más.

Important

Toolkit for Visual Studio solo está disponible para su instalación en los sistemas operativos de Windows. Si busca información sobre cómo trabajar con Visual Studio Code, consulte [AWS Toolkit for Visual Studio Code](#).

Si ha utilizado Toolkit for Visual Studio antes, probablemente ya conozca la configuración de perfiles de credenciales de AWS que contienen una clave de acceso y una clave secreta. Estos perfiles de credenciales se usan en Toolkit for Visual Studio para habilitar las llamadas a las API de servicio de AWS (por ejemplo, en Amazon S3 para obtener una lista de los buckets o en CodeCommit para obtener una lista de repositorios). Para extraer y enviar un código a un repositorio de CodeCommit, también se necesitan credenciales de Git. Si no dispone de credenciales de Git, Toolkit for Visual Studio puede generar y aplicar esas credenciales por usted. Esto puede ahorrarle mucho tiempo.

Para usar Visual Studio con CodeCommit, necesita lo siguiente:

- Un usuario de IAM con un conjunto de credenciales (una clave de acceso y una clave secreta) configurado para tal fin. Asimismo, este usuario de IAM debe tener:

Una de las políticas administradas de CodeCommit y la política administrada de IAMSelfManageServiceSpecificCredentials aplicada a esta.

O BIEN

Si el usuario de IAM ya tiene las credenciales de Git, una de las políticas administradas de CodeCommit o los permisos equivalentes.

Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit y Descripción y obtención de las credenciales de seguridad](#).

- AWS Toolkit for Visual Studio instalado en el equipo donde haya instalado Visual Studio. Para obtener más información, consulte [Configuración de AWS Toolkit for Visual Studio](#).

Para obtener más información sobre el uso de AWS Toolkit for Visual Studio con CodeCommit, consulte [Usar AWS CodeCommit con Visual Studio Team Explorer](#) en la Guía del usuario de Toolkit for Visual Studio.

Configuración para los usuarios de SSH que no utilizan AWS CLI

Si quieres usar conexiones SSH para tu repositorio, puedes conectarte a él AWS CodeCommit sin necesidad de instalar el. AWS CLI AWS CLI Incluye comandos que son útiles a la hora de utilizar y gestionar CodeCommit repositorios, pero no son necesarios para la configuración inicial.

En este tema se presupone:

- Ha configurado un usuario de IAM con las políticas o los permisos necesarios CodeCommit y la política IAMUserSSHKeysgestionada o los permisos equivalentes necesarios para cargar las claves. Para obtener más información, consulte [Uso de políticas basadas en la identidad \(políticas de IAM\) para CodeCommit](#).
- Que ya tiene, o sabe cómo crear, un par de claves pública y privada. Recomendamos encarecidamente que proteja la clave SSH con una frase de contraseña segura.
- Que está familiarizado con SSH, el cliente de Git y sus archivos de configuración.
- Que si utiliza Windows, tiene instalada una utilidad de línea de comandos, como Git Bash, que simula el shell de Bash.

Si necesita más información, siga las instrucciones que aparecen en [Para conexiones SSH en Linux, macOS o Unix](#) o [Para conexiones SSH en Windows](#).

Temas

- [Paso 1: Asociar su clave pública a su usuario de IAM](#)

- [Paso 2: CodeCommit Añádalo a tu configuración de SSH](#)
- [Sigüientes pasos](#)

Paso 1: Asociar su clave pública a su usuario de IAM

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
2. En el panel de navegación de la consola de IAM, seleccione Usuarios y, en la lista de usuarios seleccione su usuario de IAM.
3. En la pestaña Security Credentials, elija Upload SSH public key.
4. Pegue el contenido de la clave SSH pública en el campo y, a continuación, elija Upload SSH Key.

Tip

El par de claves pública y privada debe ser SSH-2 RSA, estar en formato OpenSSH y contener 2048 bits. La clave tiene un aspecto parecido al siguiente:

```
ssh-rsa EXAMPLE-
AfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJB
gNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA
5zb2x1MRIwEAYDVQQDEw1UZXRN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGFtYXp
vbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJB
gNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE user-
name@ip-192-0-2-137
```

IAM solo es compatible con claves públicas en formato OpenSSH. Si introduce la clave pública en otro formato, aparece un mensaje de error que indica que el formato de clave no es válido.

5. Copie el ID de la clave SSH (por ejemplo, *APKAEIBAERJR2EXAMPLE*) y cierre la consola.

SSH keys for AWS CodeCommit			
Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.			
Upload SSH public key			
SSH Key ID	Uploaded	Status	Actions
APKAEIBAERJR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make Inactive Show SSH Key Delete

Paso 2: CodeCommit Añádalo a tu configuración de SSH

1. En el terminal (Linux, macOS o Unix) o emulador de Bash (Windows), edite el archivo de configuración de SSH escribiendo `cat >> ~/.ssh/config`:

```
Host git-codecommit.*.amazonaws.com
User Your-SSH-Key-ID, such as APKAEIBAERJR2EXAMPLE
IdentityFile Your-Private-Key-File, such as ~/.ssh/codecommit_rsa or ~/.ssh/id_rsa
```

Tip

Si tiene más de una configuración de SSH, asegúrese de incluir las líneas en blanco antes y, a continuación, del contenido. Pulse las teclas `Ctrl` y `d` a la vez para guardar el archivo.

2. Ejecute el siguiente comando para probar la configuración SSH:

```
ssh git-codecommit.us-east-2.amazonaws.com
```

Cuando se le pida, introduzca la frase de contraseña del archivo de clave SSH. Si todo se ha configurado correctamente, deberá ver el siguiente mensaje de operación correcta:

```
You have successfully authenticated over SSH. You can use Git to interact with
CodeCommit.
```

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Para conectarse a un repositorio, siga los pasos que se detallan en [Conectarse a un repositorio](#). Para crear un repositorio, siga los pasos que se detallan en [Creación de un repositorio](#).

Pasos de configuración para las conexiones SSH a AWS CodeCommit repositorios en Linux, macOS o Unix

Antes de poder conectarte a CodeCommit por primera vez, debes completar algunos pasos de configuración iniciales. Tras configurar el ordenador y el AWS perfil, puede conectarse a un CodeCommit repositorio y clonar ese repositorio en su ordenador (lo que también se denomina creación de un repositorio local). Si Git es nuevo para usted, es posible que quiera consultar la información que aparece en [¿Dónde puedo encontrar más información sobre Git?](#).

Temas

- [Paso 1: Configuración inicial para CodeCommit](#)
- [Paso 2: Instalar Git](#)
- [Paso 3: Configurar las credenciales en Linux, macOS o Unix](#)
- [Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio](#)
- [Siguiendo pasos](#)

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para configurar una cuenta de Amazon Web Services, crear un usuario de IAM y configurar el acceso a CodeCommit.

Para crear y configurar un usuario de IAM al que acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen

expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Note

Si desea utilizar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Paso 2: Instalar Git

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a su forma de trabajar con CodeCommit ella. Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Paso 3: Configurar las credenciales en Linux, macOS o Unix

SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit

Para configurar las claves públicas y privadas de Git y CodeCommit

1. Desde el terminal de su equipo local, ejecute el comando `ssh-keygen` y siga las instrucciones para guardar el archivo en el directorio `.ssh` de su perfil.

 Note

Asegúrese de consultar al administrador del sistema dónde se deberían almacenar los archivos de las claves y qué nomenclatura de archivos debe usarse.

Por ejemplo:

```
$ ssh-keygen
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/user-name/.ssh/id_rsa): Type /home/  
your-user-name/.ssh/ and a file name here, for example /home/your-user-name/.ssh/  
codecommit_rsa
```

```
Enter passphrase (empty for no passphrase): <Type a passphrase, and then press  
Enter>
```

```
Enter same passphrase again: <Type the passphrase again, and then press Enter>
```

```
Your identification has been saved in /home/user-name/.ssh/codecommit_rsa.
```

```
Your public key has been saved in /home/user-name/.ssh/codecommit_rsa.pub.
```

```
The key fingerprint is:
```

```
45:63:d5:99:0e:99:73:50:5e:d4:b3:2d:86:4a:2c:14 user-name@client-name
The key's randomart image is:
+--[ RSA 2048]-----+
|      E.+..o*..++|
|      .o .=.o.|
|      . .. *. +|
|      ..o . +..|
|      So . . . |
|      .         |
|               |
|               |
|               |
+-----+

```

Esto genera:

- El `codecommit_rsa` archivo, que es el archivo de clave privada.
- El `codecommit_rsa.pub` archivo, que es el archivo de clave pública.

Tip

De forma predeterminada, `ssh-keygen` genera una clave de 2048 bits. Puede usar los parámetros `-t` y `-b` para especificar el tipo y la longitud de la clave. Si desea una clave de 4096 bits en formato `rsa`, debe especificarla ejecutando el comando con los siguientes parámetros:

```
ssh-keygen -t rsa -b 4096
```

Para obtener más información sobre los formatos y las longitudes necesarios para las claves SSH, consulte [Uso de IAM](#) con. CodeCommit

2. Ejecute el siguiente comando para mostrar el valor del archivo de claves públicas (`codecommit_rsa.pub`):

```
cat ~/.ssh/codecommit_rsa.pub
```

Copie este valor. Tendrá un aspecto similar al siguiente:

```
ssh-rsa EXAMPLE-AfICCQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMakGA1UEBhMCVVMxCzAJB
```

gNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxChZAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE *user-name@ip-192-0-2-137*

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.

 Note

Puede ver y administrar sus CodeCommit credenciales directamente en Mis credenciales de seguridad. Para obtener más información, consulte [Visualización y administración de sus credenciales](#).

- En el panel de navegación de la consola de IAM, seleccione Usuarios y, en la lista de usuarios seleccione su usuario de IAM.
- En la página de detalles del usuario, elija la pestaña Security Credentials y, a continuación, elija Upload SSH public key.
- Pegue el contenido de la clave SSH pública en el campo y, a continuación, elija Upload SSH public key.
- Copie o guarde la información en el identificador de clave SSH (por ejemplo, *APKAEIBAERJR2EXAMPLE*).

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. [Learn more about SSH keys.](#)

[Upload SSH public key](#)

SSH Key ID	Uploaded	Status	Actions
APKAEIBAERIR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make Inactive Show SSH Key Delete

 Note

Si has IDs cargados más de una clave SSH, las claves se muestran alfabéticamente por ID de clave, no por fecha de carga. Asegúrese de que ha copiado el ID de clave que está asociado a la fecha de carga correcta.

- En tu máquina local, usa un editor de texto para crear un archivo de configuración en el directorio `~/.ssh` y, a continuación, agrega las siguientes líneas al archivo, cuyo valor *User* es el ID de la clave SSH que copiaste anteriormente:

```
Host git-codecommit.*.amazonaws.com
  User APKAEIBAERJR2EXAMPLE
  IdentityFile ~/.ssh/codecommit_rsa
```

 Note

Si le diste a tu archivo de clave privada un nombre diferente al de «*codecommit_rsa*», asegúrate de usarlo aquí».

Puede configurar el acceso SSH a los repositorios de varias cuentas de Amazon Web Services. Para obtener más información, consulte [Solución de problemas de conexiones SSH a AWS CodeCommit](#).

Guarde este archivo y llámelo `config`.

- Desde el terminal, ejecute el siguiente comando para cambiar los permisos para el archivo de configuración:

```
chmod 600 config
```

- Ejecute el siguiente comando para probar la configuración SSH:

```
ssh git-codecommit.us-east-2.amazonaws.com
```

Se le pedirá que confirme la conexión, porque `git-codecommit.us-east-2.amazonaws.com` aún no está incluido en su archivo de hosts conocidos.

La huella digital CodeCommit del servidor se muestra como parte de la verificación (a `a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e` favor MD5 o a `31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UES56fG6ZIZQ` favor SHA256).

 Note

CodeCommit las huellas digitales del servidor son únicas para cada uno Región de AWS. Para ver las huellas digitales del servidor de un Región de AWS, consulte. [Huellas digitales del servidor para CodeCommit](#)

Una vez que haya confirmado la conexión, debería ver la confirmación que ha añadido al servidor en el archivo de hosts conocidos y un mensaje de conexión correcta. Si no ve un mensaje de confirmación, compruebe que ha guardado el config archivo en el directorio ~/.ssh del usuario de IAM al que ha configurado el acceso y que ha especificado el archivo de clave privada correcto. CodeCommit

Para obtener más información con la que solucionar problemas, ejecute el comando ssh con el parámetro -v. Por ejemplo:

```
ssh -v git-codecommit.us-east-2.amazonaws.com
```

Para obtener más información con la que solucionar problemas de conexión, consulte [Solución de problemas de conexiones SSH a AWS CodeCommit](#).

Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya le ha enviado el nombre y los detalles de conexión del repositorio de CodeCommit, puede omitir este paso y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.
 - Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
 - Copie la dirección URL HTTPS (GRC) si utiliza el comando git-remote-codecommit en su equipo local.
 - Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

 Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. Abra un terminal. Desde el directorio /tmp, ejecute el comando git clone con la URL de SSH que ha copiado para clonar el repositorio. Por ejemplo, para clonar un repositorio *MyDemoRepo* con el nombre de un repositorio local denominado *my-demo-repo* en la región EE.UU. Este (Ohio):

```
git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

 Note

Si ha probado su conexión correctamente, pero el comando de clonación devuelve un error, es posible que no tenga el acceso requerido a su archivo de configuración, o que haya otro ajuste que entre en conflicto con el archivo de configuración. Pruebe conectarse de nuevo, esta vez incluyendo el ID de la clave SSH en el comando. Por ejemplo:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Para obtener más información, consulte [Error de acceso: la clave pública se ha cargado correctamente en IAM pero se ha producido un error de conexión en los sistemas Linux, macOS o Unix](#).

Para obtener más información sobre cómo conectar con los repositorios, consulte [Conéctese al CodeCommit repositorio clonando el repositorio](#).

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Pasos de configuración para las conexiones SSH a los AWS CodeCommit repositorios en Windows

Antes de poder conectarse a AWS CodeCommit por primera vez, debe completar algunos pasos de configuración iniciales. Tras configurar el ordenador y el AWS perfil, puede conectarse a un CodeCommit repositorio y clonar ese repositorio en su ordenador (lo que también se denomina creación de un repositorio local). Si Git es nuevo para usted, es posible que quiera consultar la información que aparece en [¿Dónde puedo encontrar más información sobre Git?](#).

Temas

- [Paso 1: Configuración inicial para CodeCommit](#)
- [Paso 2: Instalar Git](#)
- [Paso 3: Configura las claves públicas y privadas para Git y CodeCommit](#)
- [Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio](#)
- [Siguientes pasos](#)

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para configurar una cuenta de Amazon Web Services, crear un usuario de IAM y configurar el acceso a CodeCommit.

Para crear y configurar un usuario de IAM al que acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

 Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

 Note

Si desea utilizar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Paso 2: Instalar Git

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y

posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Note

Git es una plataforma en evolución que se actualiza periódicamente. En ocasiones, un cambio en una función puede afectar a su funcionamiento. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Si la versión de Git que ha instalado no incluye un emulador de Bash, como, por ejemplo, Git Bash, instale uno. Utiliza este emulador en lugar de la línea de comandos de Windows para configurar las conexiones SSH.

Paso 3: Configura las claves públicas y privadas para Git y CodeCommit

Para configurar las claves públicas y privadas para Git y CodeCommit en Windows

1. Abra el emulador de Bash.

Note

Es posible que deba ejecutar el emulador con permisos administrativos.

En el emulador, ejecute el comando `ssh-keygen` y siga las instrucciones que aparecen para guardar el archivo en el directorio `.ssh` de su perfil.

Por ejemplo:

```
$ ssh-keygen
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/drive/Users/user-name/.ssh/id_rsa): Type a file name here, for example /c/Users/user-name/.ssh/codecommit_rsa
```

Enter passphrase (empty for no passphrase): *<Type a passphrase, and then press Enter>*

Enter same passphrase again: *<Type the passphrase again, and then press Enter>*

Your identification has been saved in *drive/Users/user-name/.ssh/codecommit_rsa*.

Your public key has been saved in *drive/Users/user-name/.ssh/codecommit_rsa.pub*.

The key fingerprint is:

45:63:d5:99:0e:99:73:50:5e:d4:b3:2d:86:4a:2c:14 *user-name@client-name*

The key's randomart image is:

```
+--[ RSA 2048 ]-----+
|           E.+..o*..++|
|           .o .=.o.|
|           . .. * . +|
|           ..o . +..|
|           So . . . |
|           .         |
|                   |
|                   |
|                   |
|                   |
+-----+

```

Esto genera:

- El *codecommit_rsa* archivo, que es el archivo de clave privada.
- El *codecommit_rsa* archivo.pub, que es el archivo de clave pública.

Tip

De forma predeterminada, ssh-keygen genera una clave de 2048 bits. Puede usar los parámetros -t y -b para especificar el tipo y la longitud de la clave. Si desea una clave de 4096 bits en formato rsa, debe especificarla ejecutando el comando con los siguientes parámetros:

```
ssh-keygen -t rsa -b 4096
```

Para obtener más información sobre los formatos y las longitudes necesarios para las claves SSH, consulte [Uso de IAM con](#). CodeCommit

2. Ejecute los siguientes comandos para mostrar el valor del archivo de claves públicas (*codecommit_rsa.pub*):

```
cd .ssh
notepad codecommit_rsa.pub
```

Copie el contenido del archivo y, a continuación, cierre el Bloc de notas sin guardar. El contenido del archivo es similar a este:

```
ssh-rsa EXAMPLE-AfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGxlmQ8wDQYDVQQKEwZBbWZF6b24xFDASBgNVBASTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwluZXN0Q2lsYWxHZAAdBgkqhkiG9w0BCQEWEg5vb25lQGgtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGxlmQ8wDQYDVQQKEwZBbWZF6b24xFDAS=EXAMPLE user-name@computer-name
```

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>

 Note

Puede ver y administrar sus CodeCommit credenciales directamente en Mis credenciales de seguridad. Para obtener más información, consulte [Visualización y administración de sus credenciales](#).

- En el panel de navegación de la consola de IAM, seleccione Usuarios y, en la lista de usuarios seleccione su usuario de IAM.
- En la página de detalles del usuario, elija la pestaña Security Credentials y, a continuación, elija Upload SSH public key.
- Pegue el contenido de la clave SSH pública en el campo y, a continuación, elija Upload SSH public key.
- Copie o guarde la información en el identificador de clave SSH (por ejemplo, *APKAEIBAERJR2EXAMPLE*).

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. [Learn more about SSH keys.](#)

[Upload SSH public key](#)

SSH Key ID	Uploaded	Status	Actions
APKAEI8AERIR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make Inactive Show SSH Key Delete

 Note

Si has IDs cargado más de una clave SSH, las claves se muestran alfabéticamente por ID de clave, no por fecha de carga. Asegúrese de que ha copiado el ID de clave que está asociado a la fecha de carga correcta.

8. En el emulador de Bash, ejecute los siguientes comandos para crear un archivo de configuración en el directorio `~/.ssh`, o edítelo si existiera uno:

```
notepad ~/.ssh/config
```

9. Agrega las siguientes líneas al archivo, donde el valor de *User* es el ID de clave SSH que copiaste anteriormente y el valor *IdentityFile* es la ruta y el nombre del archivo de clave privada:

```
Host git-codecommit.*.amazonaws.com
  User APKAEIBAERJR2EXAMPLE
  IdentityFile ~/.ssh/codecommit_rsa
```

 Note

Si le diste a tu archivo de clave privada un nombre diferente al *codecommit_rsa*, asegúrate de usarlo aquí.

Puede configurar el acceso SSH a los repositorios de varias cuentas de Amazon Web Services. Para obtener más información, consulte [Solución de problemas de conexiones SSH a AWS CodeCommit](#).

Guarde el archivo como `config` (no como `config.txt`) y, a continuación, cierre el Bloc de notas.

 Important

El nombre del archivo debe ser `config` sin extensión de archivo. De lo contrario, las conexiones SSH fallan.

10. Ejecute el siguiente comando para probar la configuración SSH:

```
ssh git-codecommit.us-east-2.amazonaws.com
```

Se le pedirá que confirme la conexión, porque `git-codecommit.us-east-2.amazonaws.com` aún no está incluido en su archivo de hosts conocidos. La huella digital CodeCommit del servidor se muestra como parte de la verificación (a `a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e` favor MD5 o a `31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UES56fG6ZIzQ` favor SHA256).

Note

CodeCommit las huellas digitales del servidor son únicas para cada uno Región de AWS. Para ver las huellas digitales del servidor de un Región de AWS, consulte [Huellas digitales del servidor para CodeCommit](#)

Una vez que haya confirmado la conexión, debería ver la confirmación que ha añadido al servidor en el archivo de hosts conocidos y un mensaje de conexión correcta. Si no ve un mensaje de confirmación, compruebe que ha guardado el `config` archivo en el directorio `~/.ssh` del usuario de IAM al que ha configurado el acceso CodeCommit, que el `config` archivo no tiene extensión de archivo (por ejemplo, no debe llamarse `config.txt`) y que ha especificado el archivo de clave privada correcto (no `.pub`). `codecommit_rsa codecommit_rsa`

Para solucionar problemas, ejecute el comando `ssh` con el parámetro `-v`. Por ejemplo:

```
ssh -v git-codecommit.us-east-2.amazonaws.com
```

Para obtener más información con la que solucionar problemas de conexión, consulte [Solución de problemas de conexiones SSH a AWS CodeCommit](#).

Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya le ha enviado el nombre y los detalles de conexión del repositorio de CodeCommit, puede omitir este paso y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.
 - Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
 - Copie la dirección URL HTTPS (GRC) si utiliza el comando git-remote-codecommit en su equipo local.
 - Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

 Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. En el emulador de Bash, ejecute el comando git clone con la URL SSH que copió para clonar el repositorio. Este comando crea el repositorio local en un subdirectorio del directorio en el que ejecuta el comando. Por ejemplo, para clonar un repositorio *MyDemoRepo* con el nombre de un repositorio local denominado *my-demo-repo* en la región EE.UU. Este (Ohio):

```
git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

También puede abrir un símbolo del sistema y, con la URL y el ID de clave SSH de la clave pública que ha cargado en IAM, ejecutar el comando git clone. El repositorio local se crea en un subdirectorio del directorio en el que ejecuta el comando. Por ejemplo, para clonar un repositorio denominado *MyDemoRepo* en un repositorio local llamado *my-demo-repo*:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Para obtener más información, consulte [Conéctese al CodeCommit repositorio clonando el repositorio](#) y [Creación de una confirmación](#).

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Pasos de configuración para las conexiones HTTPS a AWS CodeCommit repositorios en Linux, macOS o Unix con el AWS CLI asistente de credenciales

Antes de poder conectarte a AWS CodeCommit por primera vez, debes completar los pasos de configuración iniciales. Para la mayoría de usuarios, se puede realizar fácilmente siguiendo los pasos que se detallan en [Para usuarios HTTPS mediante credenciales de Git](#). Sin embargo, si desea conectarse CodeCommit mediante una cuenta raíz, un acceso federado o credenciales temporales, puede utilizar el asistente de credenciales que se incluye en. AWS CLI

Note

Si bien el asistente de credenciales es un método compatible para conectarse CodeCommit mediante un acceso federado, un proveedor de identidad o credenciales temporales, el método recomendado es instalar y usar la utilidad. `git-remote-codecommit` Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Temas

- [Paso 1: Configuración inicial para CodeCommit](#)
- [Paso 2: Instalar Git](#)
- [Paso 3: Configurar el ayudante de credenciales](#)
- [Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio](#)
- [Siguientes pasos](#)

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para configurar una cuenta de Amazon Web Services, crear y configurar un usuario de IAM e instalar la AWS CLI.

Para crear y configurar un usuario de IAM para acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Para instalar y configurar AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con él CodeCommit desde la línea de comandos. Le recomendamos que instale la versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales congit-remote-codecommit.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

 Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplo `us-east-2`. Cuando se le

pregunte el formato de salida predeterminado, indique json. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
Default region name [None]: Type a supported region for CodeCommit here, and then press Enter
Default output format [None]: Type json here, and then press Enter
```

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)
- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o a un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- ~~sa-east-1~~

- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1
- ca-central-1
- us-gov-west-1
- us-gov-east-1
- eu-north-1
- ap-east-1
- me-south-1
- cn-north-1
- cn-northwest-1
- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

Paso 2: Instalar Git

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a su forma de trabajar con CodeCommit ella. Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Paso 3: Configurar el ayudante de credenciales

1. Desde el terminal, usa Git para ejecutar `git config`, especificando el uso del ayudante de credenciales de Git con el perfil de AWS credenciales y habilitando el ayudante de credenciales de Git para enviar la ruta a los repositorios:

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

 Tip

El ayudante de credenciales usa el perfil de AWS credenciales predeterminado o el rol de instancia de Amazon EC2. Puede especificar el perfil que se va a utilizar, como `CodeCommitProfile`, si ha creado un perfil de credenciales de AWS para utilizarse en CodeCommit:

```
git config --global credential.helper '!aws --profile CodeCommitProfile
codecommit credential-helper $@'
```

Si el nombre del perfil contiene espacios, asegúrese de incluir el nombre entre comillas ("").

Puede configurar perfiles por repositorio en lugar de hacerlo a nivel global. Para ello, utilice `--local` en lugar de `--global`.

El auxiliar de credenciales de Git escribe el siguiente valor en `~/.gitconfig`:

```
[credential]
  helper = !aws --profile CodeCommitProfile codecommit credential-helper $@
```

```
UseHttpPath = true
```

Important

Si desea utilizar un usuario de IAM diferente en la misma máquina local CodeCommit, debe volver a ejecutar el git config comando y especificar un perfil de credenciales diferente. AWS

2. Ejecute git config --global --edit para comprobar que el valor anterior se ha escrito en ~/.gitconfig. Si se ejecuta correctamente, debería ver el valor anterior (además de otros valores que podrían existir en el archivo de configuración global de Git). Para salir, escriba :q y, a continuación, pulse Intro.

Si experimenta problemas después de configurar el auxiliar de credenciales, consulte [Solución de problemas](#).

Important

Si es usuario de macOS, siga estos pasos para comprobar que el ayudante de credenciales está configurado correctamente.

3. Si utilizas macOS, usa HTTPS para [conectarte a un CodeCommit repositorio](#). Después de conectarte a un CodeCommit repositorio con HTTPS por primera vez, el acceso posterior fallará transcurridos unos 15 minutos. La versión de Git predeterminada en macOS utiliza Keychain Access para almacenar las credenciales. Por motivos de seguridad, la contraseña generada para acceder al CodeCommit repositorio es temporal, por lo que las credenciales almacenadas en el llavero dejan de funcionar transcurridos unos 15 minutos. Para evitar el uso de credenciales caducadas, deberá:
 - Instalar una versión de Git que no utilice la cadena de claves de forma predeterminada, o bien
 - Configure la utilidad Keychain Access para que no proporcione credenciales a los repositorios. CodeCommit
1. Abra la utilidad Keychain Access. (Puede utilizar la función Finder para localizarlo).
 2. Busque la opción git-codecommit.us-east-2.amazonaws.com. Marque la fila, abra el menú contextual o haga clic con el botón derecho en ella y, a continuación, elija Get Info.
 3. Elija la pestaña Access Control.

4. En Confirm before allowing access, elija `git-credential-osxkeychain` y, a continuación, seleccione el signo menos para eliminarla de la lista.

 Note

Una vez que haya eliminado `git-credential-osxkeychain` de la lista, ve un mensaje emergente cada vez que ejecute un comando de Git. Elija Deny para continuar. Si las ventanas emergentes le interrumpen demasiado, estas son otras opciones:

- Conéctese para CodeCommit usar SSH en lugar de HTTPS. Para obtener más información, consulte [Para conexiones SSH en Linux, macOS o Unix](#).
- En la utilidad Keychain Access, en la pestaña Access Control de `git-codecommit.us-east-2.amazonaws.com`, elija la opción Allow all applications to access this item (access to this item is not restricted). Esto impide que aparezcan ventanas emergentes, pero las credenciales acabarán caducando (de media, duran unos 15 minutos) y aparece un mensaje de error 403. Cuando suceda esto, elimine el elemento de la cadena de claves para restaurar la funcionalidad.
- Para obtener más información, consulte [Git para macOS: he configurado el ayudante de credenciales correctamente, pero ahora no tengo acceso a mi repositorio \(403\)](#).

Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya le ha enviado el nombre y los detalles de conexión del CodeCommit repositorio, puede omitir este paso y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.

- Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
- Copie la dirección URL HTTPS (GRC) si utiliza el comando `git-remote-codecommit` en su equipo local.
- Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. Abra un terminal y ejecute el comando `git clone` con la URL HTTPS que ha copiado. Por ejemplo, para clonar un repositorio *MyDemoRepo* con el nombre de un repositorio local denominado *my-demo-repo* en la región EE.UU. Este (Ohio):

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Pasos de configuración para las conexiones HTTPS a AWS CodeCommit los repositorios en Windows con el asistente de AWS CLI credenciales

Antes de poder conectarse a AWS CodeCommit por primera vez, debe completar los pasos de configuración iniciales. Para la mayoría de usuarios, se puede realizar fácilmente siguiendo los pasos que se detallan en [Para usuarios HTTPS mediante credenciales de Git](#). Sin embargo, si desea

conectarse CodeCommit mediante una cuenta raíz, un acceso federado o credenciales temporales, puede utilizar el asistente de credenciales que se incluye en. AWS CLI

 Note

Si bien el asistente de credenciales es un método compatible para conectarse CodeCommit mediante un acceso federado, un proveedor de identidad o credenciales temporales, el método recomendado es instalar y usar la utilidad. `git-remote-codecommit` Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

En este tema se explican los pasos para instalar AWS CLI, configurar el ordenador y el AWS perfil, conectarse a un CodeCommit repositorio y clonar ese repositorio en el ordenador, lo que también se conoce como creación de un repositorio local. Si Git es nuevo para usted, es posible que quiera consultar la información que aparece en [¿Dónde puedo encontrar más información sobre Git?](#).

Temas

- [Paso 1: Configuración inicial para CodeCommit](#)
- [Paso 2: Instalar Git](#)
- [Paso 3: Configurar el ayudante de credenciales](#)
- [Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio](#)
- [Siguiendo pasos](#)

Paso 1: Configuración inicial para CodeCommit

Siga estos pasos para configurar una cuenta de Amazon Web Services, crear y configurar un usuario de IAM e instalar la AWS CLI. AWS CLI Incluye un asistente de credenciales que puede configurar para las conexiones HTTPS a sus CodeCommit repositorios.

Para crear y configurar un usuario de IAM para acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario

de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

 Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Para instalar y configurar AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con él CodeCommit desde la línea de comandos. Le recomendamos que instale la versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales congint-remote-codecommit.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

 Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplous-east-2. Cuando se le pregunte el formato de salida predeterminado, indique json. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
Default region name [None]: Type a supported region for CodeCommit here, and then press Enter
Default output format [None]: Type json here, and then press Enter
```

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)

- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o a un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- sa-east-1
- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1
- ca-central-1
- us-gov-west-1
- us-gov-east-1
- eu-north-1
- ap-east-1
- me-south-1
- cn-north-1

- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

Paso 2: Instalar Git

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git for Windows](#). Si utiliza este enlace para instalar Git, puede aceptar todos los ajustes predeterminados de la instalación excepto los siguientes:

- Cuando se le pida durante el paso Ajustar su entorno PATH, seleccione la opción de usar Git desde la línea de comandos.
- (Opcional) Si pretendes usar HTTPS con el asistente de credenciales que viene incluido en el en AWS CLI lugar de configurar las credenciales de Git para CodeCommit, en la página Configuración de opciones adicionales, asegúrate de que la opción Habilitar el administrador de credenciales de Git esté desactivada. El administrador de credenciales de Git solo es compatible CodeCommit si los usuarios de IAM configuran las credenciales de Git. Para obtener más información, consulte [Para usuarios HTTPS mediante credenciales de Git](#) y [Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio \(403\)](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a la forma en que funciona. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

Paso 3: Configurar el ayudante de credenciales

AWS CLI Incluye un ayudante de credenciales de Git con el que puedes usarlo. CodeCommit El asistente de credenciales de Git requiere un perfil de AWS credenciales, que almacena una copia del ID de clave de acceso y la clave de AWS acceso AWS secreta de un usuario de IAM (junto con un Región de AWS nombre y un formato de salida predeterminados). El asistente de credenciales de Git utiliza esta información para autenticarse automáticamente, por CodeCommit lo que no es necesario que introduzcas esta información cada vez que utilices Git para interactuar con él. CodeCommit

1. Abre una línea de comandos y usa Git para ejecutar `git config`, especificando el uso del ayudante de credenciales de Git con el perfil de AWS credenciales, que permite al ayudante de credenciales de Git enviar la ruta a los repositorios:

```
git config --global credential.helper "!aws codecommit credential-helper $@"  
git config --global credential.UseHttpPath true
```

El auxiliar de credenciales de Git escribe lo siguiente en el archivo `.gitconfig`:

```
[credential]  
  helper = !aws codecommit credential-helper $@  
  UseHttpPath = true
```

 Important

- Si utiliza un emulador de Bash en lugar de la línea de comandos de Windows, debe utilizar comillas simples en lugar de comillas dobles.
- El ayudante de credenciales usa el AWS perfil predeterminado o el rol de instancia de Amazon EC2. Si ha creado un perfil de AWS credenciales para usarlo, por

ejemplo *CodeCommitProfile*, puede modificar el comando de la siguiente manera para usarlo en su lugar:

```
git config --global credential.helper "!aws codecommit credential-helper  
--profile CodeCommitProfile @"
```

Al hacerlo, se escribe lo siguiente en el archivo `.gitconfig`:

```
[credential]  
  helper = !aws codecommit credential-helper --profile=CodeCommitProfile  
  @$@  
  UseHttpPath = true
```

- Si su nombre de perfil contiene espacios, deberá editar el archivo `.gitconfig` después de ejecutar este comando para definirlo entre comillas simples (`'`). De lo contrario, el auxiliar de credenciales no funcionará.
- Si la instalación de Git para Windows incluye el Administrador de credenciales de Git, ve errores 403 o instrucciones para proporcionar credenciales en el Administrador de credenciales después de los primeros intentos de conexión. El método más fiable para resolver este problema es desinstalar y reinstalar Git para Windows sin la opción del Administrador de credenciales de Git, ya que no es compatible con CodeCommit. Si quieres conservar la utilidad Git Credential Manager, debes realizar pasos de configuración adicionales para utilizarla también CodeCommit, incluida la modificación manual del archivo `.gitconfig` para especificar el uso del ayudante de credenciales al conectarte a AWS CodeCommit. Elimine todas las credenciales almacenadas en el Administrador de credenciales (puede encontrar esta utilidad en el Panel de control). Una vez que haya eliminado las credenciales almacenadas, añada las siguientes a su archivo `.gitconfig`, guárdelo y, a continuación, intente conectarse de nuevo desde una ventana de símbolo del sistema:

```
[credential "https://git-codecommit.us-east-2.amazonaws.com"]  
  helper = !aws codecommit credential-helper @$@  
  UseHttpPath = true  
  
[credential "https://git-codecommit.us-east-1.amazonaws.com"]  
  helper = !aws codecommit credential-helper @$@  
  UseHttpPath = true
```

Es posible que tenga que volver a configurar los valores de git config especificando `--system` en lugar de `--global` o `--local` para que todas las conexiones funcionen según lo previsto.

- Si desea utilizar distintos usuarios de IAM en la misma máquina local CodeCommit, debe especificar y ejecutar la configuración `git config --local` para cada git config --global perfil de credenciales. AWS

2. Ejecute `git config --global --edit` para verificar que los valores anteriores se han escrito en el archivo `.gitconfig` para su perfil de usuario (de forma predeterminada, `%HOME%\.gitconfig` o `drive:\Users\UserName\.gitconfig`). Si se ejecuta correctamente, debería ver los valores anteriores (además de otros valores que podrían existir en el archivo de configuración global de Git). Para salir, escriba `:q` y, a continuación, pulse Intro.

Paso 4: Conectarse a la CodeCommit consola y clonar el repositorio

Si un administrador ya le ha enviado el nombre y los detalles de conexión del CodeCommit repositorio, puede omitir este paso y clonar el repositorio directamente.

Para conectarse a un CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.
 - Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.
 - Copie la dirección URL HTTPS (GRC) si utiliza el comando `git-remote-codecommit` en su equipo local.
 - Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

 Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. Abra un símbolo del sistema y ejecute el comando `git clone` con la URL HTTPS que ha copiado. El repositorio local se crea en un subdirectorío del directorio en el que ejecuta el comando. Por ejemplo, para clonar un repositorio *MyDemoRepo* con el nombre de un repositorio local denominado *my-demo-repo* en la región EE.UU. Este (Ohio):

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

En algunas versiones de Windows, es posible que aparezca un mensaje emergente que solicite su nombre de usuario y contraseña. Este es el sistema de gestión de credenciales integrado para Windows, pero no es compatible con el auxiliar de credenciales de AWS CodeCommit. Elija Cancelar.

Siguientes pasos

Ha completado los requisitos previos. Sigue los pasos que se indican [Empezando con CodeCommit](#) para empezar a usarlo CodeCommit.

Introducción

La forma más sencilla de empezar CodeCommit es seguir los pasos que se indican en [Empezando con CodeCommit](#). Si eres nuevo en Git CodeCommit, también deberías considerar seguir los pasos que se indican en [Primeros pasos con Git y CodeCommit](#). Esto te ayudará a familiarizarte con CodeCommit los aspectos básicos del uso de Git cuando interactúes con tus CodeCommit repositorios.

También puedes seguir el tutorial de [Simple Pipeline Walkthrough CodePipeline y](#) aprender CodeCommit a usar tu CodeCommit repositorio como parte de un proceso de entrega continua.

Los tutoriales de esta sección se han redactado desde el supuesto de que ha completado los [requisitos previos y la configuración](#), lo que incluye:

- Asignar permisos al usuario de IAM.
- Configurar la gestión de credenciales para las conexiones HTTPS o SSH en el equipo local que está utilizando para este tutorial
- Configurar AWS CLI si desea utilizar la línea de comandos o el terminal para todas las operaciones, incluida la creación del repositorio.

Temas

- [Empezar con AWS CodeCommit](#)
- [Primeros pasos con Git y AWS CodeCommit](#)

Empezar con AWS CodeCommit

En este tutorial se muestra cómo utilizar algunas CodeCommit funciones clave. Primero, creará un repositorio y confirmará algunos cambios en él. A continuación, examinará los archivos y verá los cambios. También puede crear una solicitud de extracción para que otros usuarios puedan revisar y comentar los cambios realizados en el código.

Si desea migrar el código existente a CodeCommit, consulte [Migración a AWS CodeCommit](#).

Si no está familiarizado con Git, puede ser conveniente realizar también [Primeros pasos con Git y CodeCommit](#). Tras completar estos tutoriales, deberías tener suficiente práctica CodeCommit para empezar a utilizarlos en tus propios proyectos y en entornos de equipo.

La CodeCommit consola incluye información útil en un panel plegable que puede abrir desde el icono de información



o desde cualquier enlace de información de la página. Puede cerrar este panel en cualquier momento.

La imagen muestra la interfaz de usuario de la consola de AWS CodeCommit. En la parte superior, hay una barra de navegación con los enlaces: Developer Tools > CodeCommit > Repositories > MyDemoRepo. Debajo, el título 'MyDemoRepo' está seguido de botones para 'Notify', 'main' (seleccionado), 'Create pull request' y 'Clone URL'. El área principal muestra el código de un archivo llamado 'cl_sample.js' con una barra de búsqueda y un botón 'Edit'. El código es un script de Node.js que utiliza el SDK de AWS para interactuar con CodeCommit. A la derecha, un panel plegable titulado 'View a file in a repository' contiene texto explicativo sobre cómo ver y editar archivos, una advertencia sobre los límites de tamaño y una lista de enlaces para 'Browse files', 'Edit a file' y 'Add a file'.

La CodeCommit consola también te permite buscar rápidamente tus recursos, como repositorios, proyectos de compilación, aplicaciones de despliegue y canalizaciones. Elija Ir a recurso o pulse la tecla / y, a continuación, escriba el nombre del recurso. Se muestran todas las coincidencias en la lista. En las búsquedas, no se distingue entre mayúsculas y minúsculas. Solo puede ver los recursos para los que tiene permiso. Para obtener más información, consulte [Visualización de recursos en la consola](#).

Requisitos previos

Antes de comenzar, debe completar el procedimiento de [requisitos previos y configuración](#), que incluye:

- Asignar permisos al usuario de IAM.
- Configurar la administración de credenciales para las conexiones HTTPS o SSH en el equipo local que utilice para este tutorial.
- Configurar AWS CLI si desea utilizar la línea de comandos o el terminal para todas las operaciones, incluida la creación del repositorio.

Temas

- [Paso 1: Crear un CodeCommit repositorio](#)
- [Paso 2: Añadir archivos a su repositorio](#)
- [Paso 3: Examinar el contenido del repositorio](#)
- [Paso 4: Crear y colaborar en una solicitud de extracción](#)
- [Paso 5: Eliminar](#)
- [Paso 6: Siguiendo pasos](#)

Paso 1: Crear un CodeCommit repositorio

Puede usar la CodeCommit consola para crear un CodeCommit repositorio. Si ya tiene un repositorio y desea utilizarlo en este tutorial, puede omitir este paso.

Note

Según el uso, es posible que se le cobre por crear un repositorio o por acceder a él. Para obtener más información, consulta [los precios](#) en la página de información CodeCommit del producto.

Para crear el CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Usa el selector de regiones para elegir Región de AWS dónde quieres crear el repositorio. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositorios, seleccione Crear repositorio.
4. En la página Create repository (Crear repositorio), en Repository name (Nombre del repositorio), escriba un nombre para el repositorio (por ejemplo, **MyDemoRepo**).

 Note

En los nombres de los repositorios se distinguen mayúsculas de minúsculas y no pueden tener más de 100 caracteres. Para obtener más información, consulte [Límites](#).

5. (Opcional) En Description (Descripción), introduzca una descripción (por ejemplo, **My demonstration repository**). Esto puede ayudarle a usted y a otros usuarios a conocer el propósito del repositorio.
6. (Opcional) Seleccione Añadir etiqueta para añadir una o más etiquetas de repositorio (una etiqueta de atributo personalizada que le ayuda a organizar y gestionar sus AWS recursos) a su repositorio. Para obtener más información, consulte [Etiquetar repositorios en AWS CodeCommit](#).
7. (Opcional) Amplíe la configuración adicional para especificar si desea utilizar la clave predeterminada Clave administrada de AWS o la propia clave gestionada por el cliente para cifrar y descifrar los datos de este repositorio. Si decide utilizar su propia clave gestionada por el cliente, debe asegurarse de que esté disponible en el Región de AWS lugar donde vaya a crear el repositorio y de que la clave esté activa. Para obtener más información, consulte [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#).
8. (Opcional) Seleccione Habilitar Amazon CodeGuru Reviewer para Java y Python si este repositorio va a contener código de Java o Python y desea que CodeGuru Reviewer analice ese código. CodeGuru Reviewer utiliza varios modelos de aprendizaje automático para detectar defectos en el código y sugerir automáticamente mejoras y correcciones en las solicitudes de extracción de cambios. Para obtener más información, consulte la Guía del usuario de Amazon CodeGuru Reviewer.
9. Seleccione Crear.

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

Tags

☐ **Enable Amazon CodeGuru Reviewer for Java and Python - *optional***

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

[Cancel](#)[Create](#)

Note

Si utiliza un nombre de repositorio que no sea MyDemoRepo, asegúrese de utilizarlo en los demás pasos.

Cuando se abra el repositorio, verá información sobre cómo añadir archivos directamente desde la CodeCommit consola.

Paso 2: Añadir archivos a su repositorio

Puede añadir archivos a su repositorio de las siguientes formas:

- Crear un archivo en la CodeCommit consola. Si crea el primer archivo para un repositorio en la consola, se creará una ramificación llamada principal. Esta ramificación es la ramificación predeterminada de su repositorio.
- Cargar un archivo desde el ordenador local mediante la CodeCommit consola. Si sube el primer archivo de un repositorio desde la consola, se crea una ramificación para usted llamada principal. Esta ramificación es la ramificación predeterminada de su repositorio.
- Utilizar un cliente Git para clonar el repositorio en tu ordenador local y, a continuación, añadir, confirmar y enviar archivos al CodeCommit repositorio. Se crea una ramificación para usted como parte de esa primera confirmación desde Git y se establece como la ramificación predeterminada para su repositorio. El nombre de la ramificación es la opción predeterminada de su cliente de Git. Considere configurar su cliente de Git para que use principal como nombre de la ramificación inicial.

Note

Puede crear ramificaciones y cambiar la ramificación predeterminada de un repositorio en cualquier momento. Para obtener más información, consulte [Trabajando con ramas en AWS CodeCommit repositorios](#).

La forma más sencilla de empezar es abrir la CodeCommit consola y añadir un archivo. Al hacerlo, también creará una ramificación predeterminada para su repositorio llamada principal. Para obtener instrucciones sobre cómo añadir un archivo y crear una primera confirmación en un repositorio mediante el AWS CLI, consulte [Crear la primera confirmación para un repositorio mediante el AWS CLI](#).

Para añadir un archivo a un repositorio.

1. En la barra de navegación del repositorio, elija Code (Código).
2. Elija Add file (Añadir archivo) y, a continuación, elija si desea crear o cargar un archivo desde su equipo. En este tutorial se muestra cómo hacer ambas cosas.
3. Para añadir un archivo, haga lo siguiente:

- a. En la lista desplegable de ramificaciones, elija la ramificación donde desea agregar el archivo. La ramificación predeterminada es la que se selecciona automáticamente. En el ejemplo que se muestra aquí, se denomina a la rama predeterminada *main*. Si desea agregar el archivo a otra ramificación, elija una ramificación diferente.
- b. En File name (Nombre de archivo), escriba un nombre para el archivo. En el editor de código, escriba el código del archivo.
- c. En Author name (Nombre del autor), introduzca el nombre que desea que vean los demás usuarios del repositorio.
- d. En Email address (Dirección de correo electrónico), introduzca una dirección de correo electrónico.
- e. (Opcional) En Commit message (Mensaje de confirmación), introduzca un breve mensaje. Aunque es opcional, le recomendamos que añada un mensaje de confirmación para ayudar a los miembros del equipo a entender por qué añadió este archivo. Si no introduce ningún mensaje de confirmación, se utiliza un mensaje predeterminado.
- f. Seleccione Confirmar cambios.

Para cargar un archivo, haga lo siguiente:

- Si carga un archivo, elija el archivo que desea cargar.

Upload a file

MyDemoRepo [Info](#)

Name	Size	Actions
Upload file Choose a file to upload. Choose file		

Commit changes to master

Author name

Email address

Commit message - optional
A default commit message will be used if you do not provide one.

[Cancel](#) [Commit changes](#)

- En Author name (Nombre del autor), introduzca el nombre que desea que vean los demás usuarios del repositorio.
- En Email address (Dirección de correo electrónico), introduzca una dirección de correo electrónico.
- (Opcional) En Commit message (Mensaje de confirmación), introduzca un breve mensaje. Aunque es opcional, le recomendamos que añada un mensaje de confirmación para ayudar a los miembros del equipo a entender por qué añadió este archivo. Si no introduce ningún mensaje de confirmación, se utiliza un mensaje predeterminado.
- Seleccione Confirmar cambios.

Para obtener más información, consulte [Trabajar con archivos en repositorios de AWS CodeCommit](#).

Para usar un cliente Git para clonar el repositorio, instala Git en tu equipo local y, a continuación, clona el CodeCommit repositorio. Añada algunos archivos al repositorio local y envíelos al repositorio de CodeCommit . Para obtener una introducción más detallada, consulte [Primeros pasos con Git y CodeCommit](#). Si estás familiarizado con Git, pero no estás seguro de cómo hacerlo con un CodeCommit repositorio, puedes ver ejemplos e instrucciones en [Creación de una confirmación](#) [Paso 2: Crear un repositorio local](#), o [Conectarse a un repositorio](#).

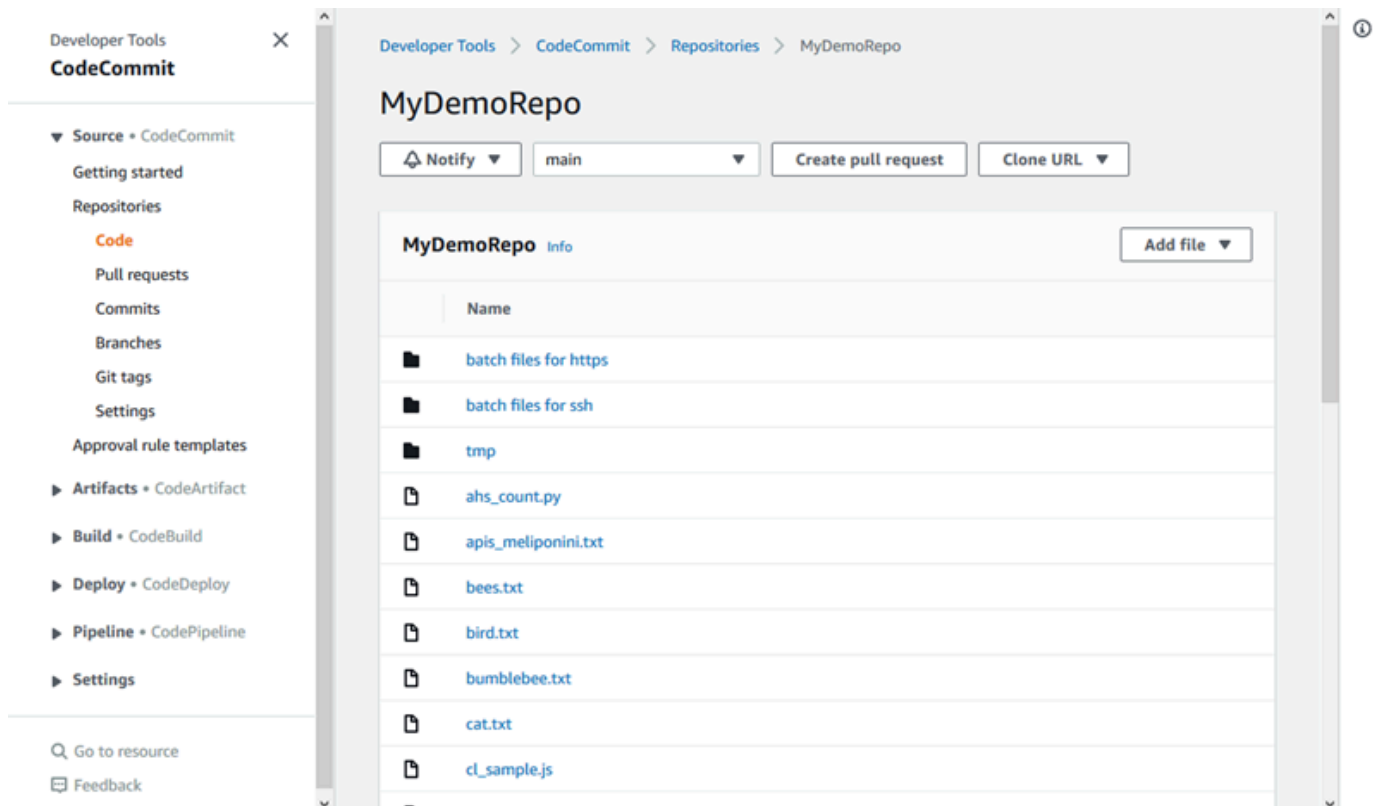
Una vez que hayas añadido algunos archivos al CodeCommit repositorio, podrás verlos en la consola.

Paso 3: Examinar el contenido del repositorio

Puede utilizar la CodeCommit consola para revisar los archivos de un repositorio o leer rápidamente el contenido de un archivo. Esto le ayuda a determinar qué ramificación comprobar o si crear una copia local de un repositorio.

Para examinar el repositorio

1. En Repositorios, selecciona MyDemoRepo.
2. La página muestra el contenido de la ramificación predeterminada del repositorio. Para ver otra ramificación o para ver el código en una etiqueta específica, elija en la lista la ramificación o la etiqueta que desea ver. En la siguiente captura de pantalla, la vista se establece en la ramificación principal.



Developer Tools > CodeCommit > Repositories > MyDemoRepo

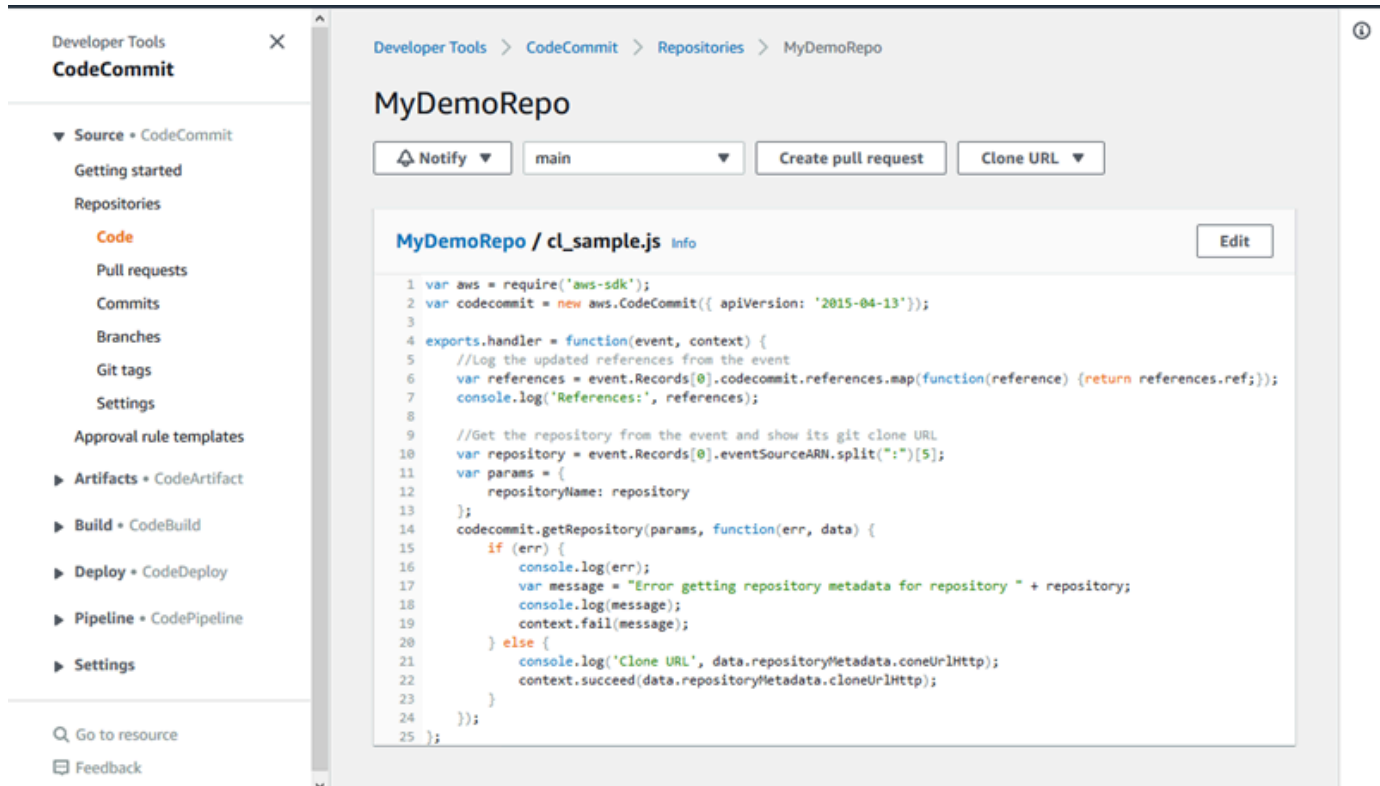
MyDemoRepo

Notify main Create pull request Clone URL

MyDemoRepo Info Add file

Name
batch files for https
batch files for ssh
tmp
ahs_count.py
apis_meliponini.txt
bees.txt
bird.txt
bumblebee.txt
cat.txt
cl_sample.js

- Para ver el contenido de un archivo del repositorio, elija el archivo en la lista. Para cambiar el color del código mostrado, elija el icono de configuración.



Developer Tools > CodeCommit > Repositories > MyDemoRepo

MyDemoRepo

Notify main Create pull request Clone URL

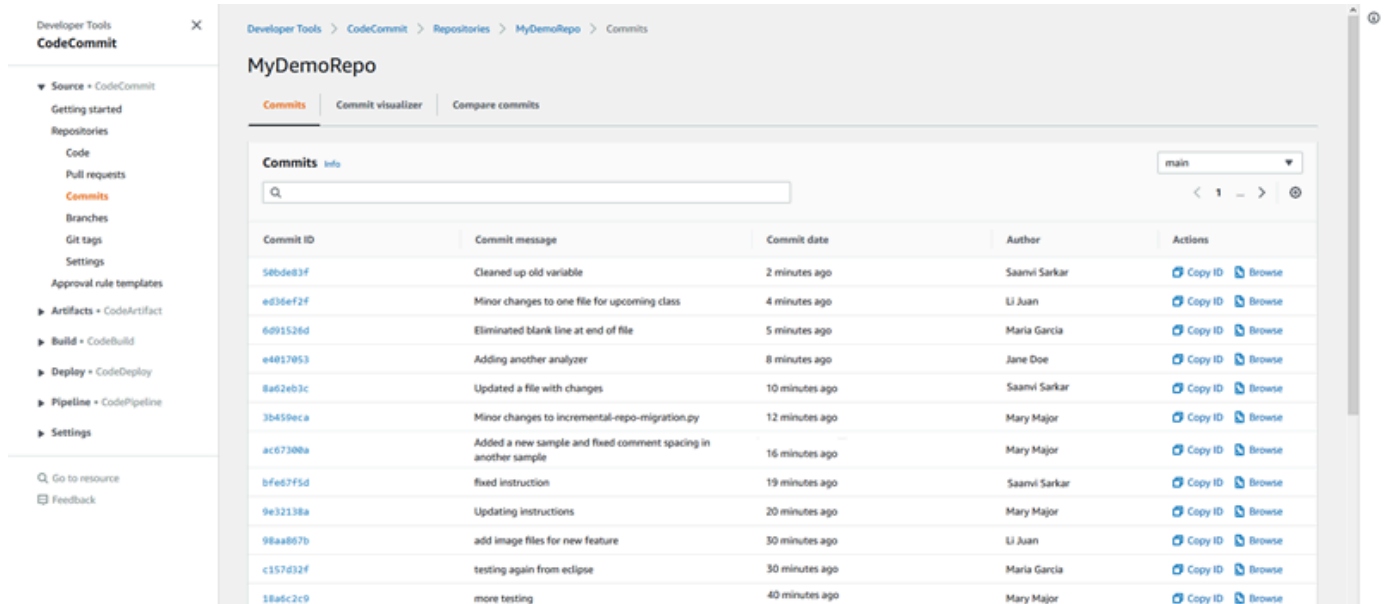
MyDemoRepo / cl_sample.js Info Edit

```

1 var aws = require('aws-sdk');
2 var codecommit = new aws.CodeCommit({ apiVersion: '2015-04-13' });
3
4 exports.handler = function(event, context) {
5   //Log the updated references from the event
6   var references = event.Records[0].codecommit.references.map(function(reference) {return reference.ref;});
7   console.log('References:', references);
8
9   //Get the repository from the event and show its git clone URL
10  var repository = event.Records[0].eventSourceARN.split(':')[5];
11  var params = {
12    repositoryName: repository
13  };
14  codecommit.getRepository(params, function(err, data) {
15    if (err) {
16      console.log(err);
17      var message = "Error getting repository metadata for repository " + repository;
18      console.log(message);
19      context.fail(message);
20    } else {
21      console.log('Clone URL', data.repositoryMetadata.cloneUrlHttp);
22      context.succeed(data.repositoryMetadata.cloneUrlHttp);
23    }
24  });
25 };
```

Para obtener más información, consulte [Navegación por los archivos en un repositorio](#).

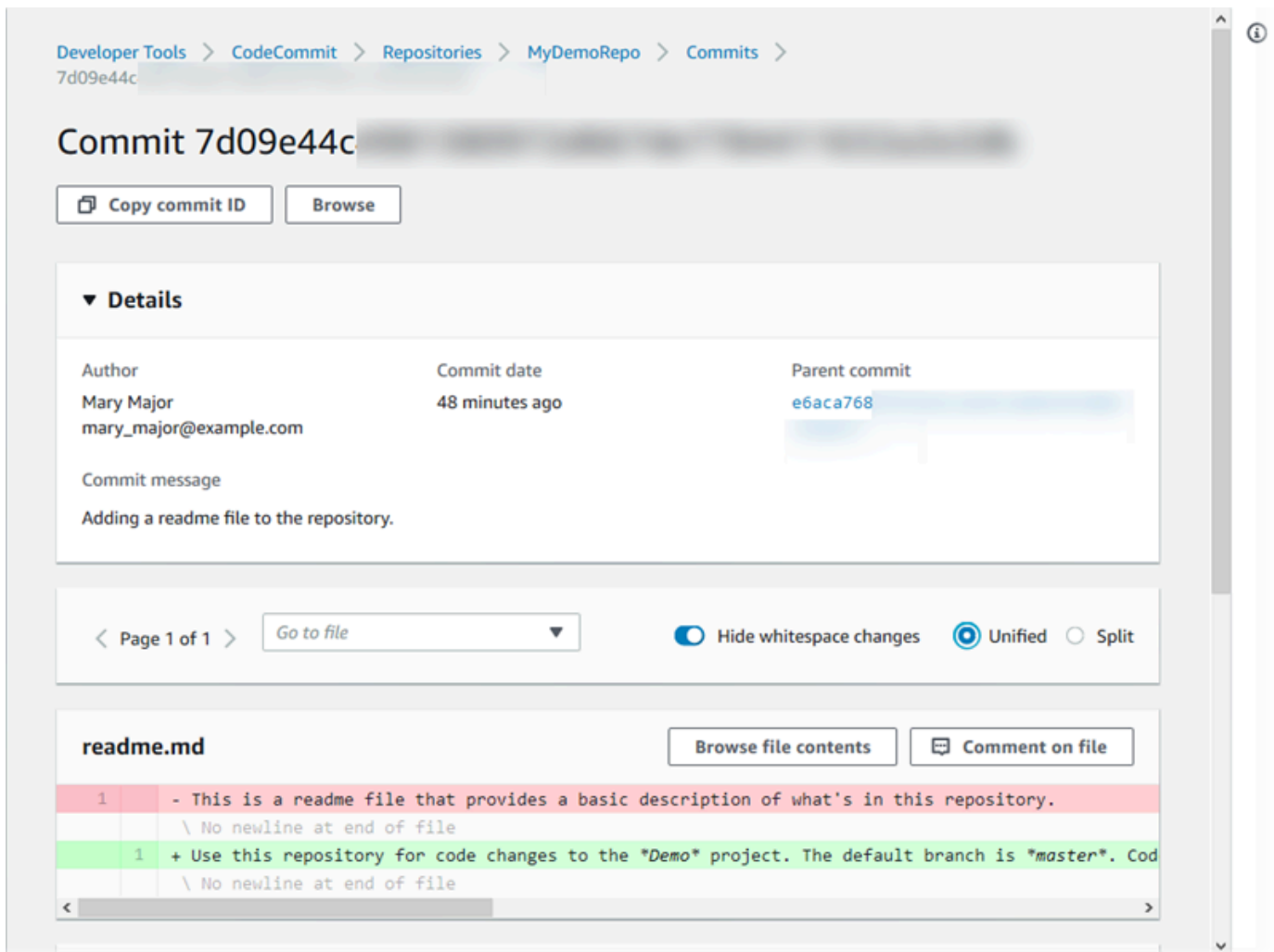
- Para examinar el historial de confirmaciones del repositorio, seleccione Commits (Confirmaciones). La consola muestra el historial de confirmaciones de la ramificación predeterminada, en orden cronológico inverso. Puede revisar los detalles de las confirmaciones por autor, fecha y otros criterios.



- Para ver el historial de confirmaciones por [ramificación](#) o por [etiqueta Git](#), elija en la lista la ramificación o la etiqueta que desea ver.
- Para ver las diferencias entre una confirmación y su confirmación primaria, elija el ID de confirmación abreviado. Puede elegir cómo mostrar los cambios, por ejemplo mostrando u ocultando los cambios en los espacios en blanco, y elegir si desea ver los cambios en línea (vista Unified [Unificada]) o de forma paralela (vista Split [Dividida]).

Note

Las preferencias para ver el código y otra configuración de la consola se guardan como cookies del navegador siempre que se modifican. Para obtener más información, consulte [Trabajo con las preferencias del usuario](#).



7. Para ver todos los comentarios de una confirmación, seleccione la confirmación y, a continuación, desplácese por los cambios para verlos en línea. También puede añadir sus propios comentarios y responder a los comentarios de otras personas.

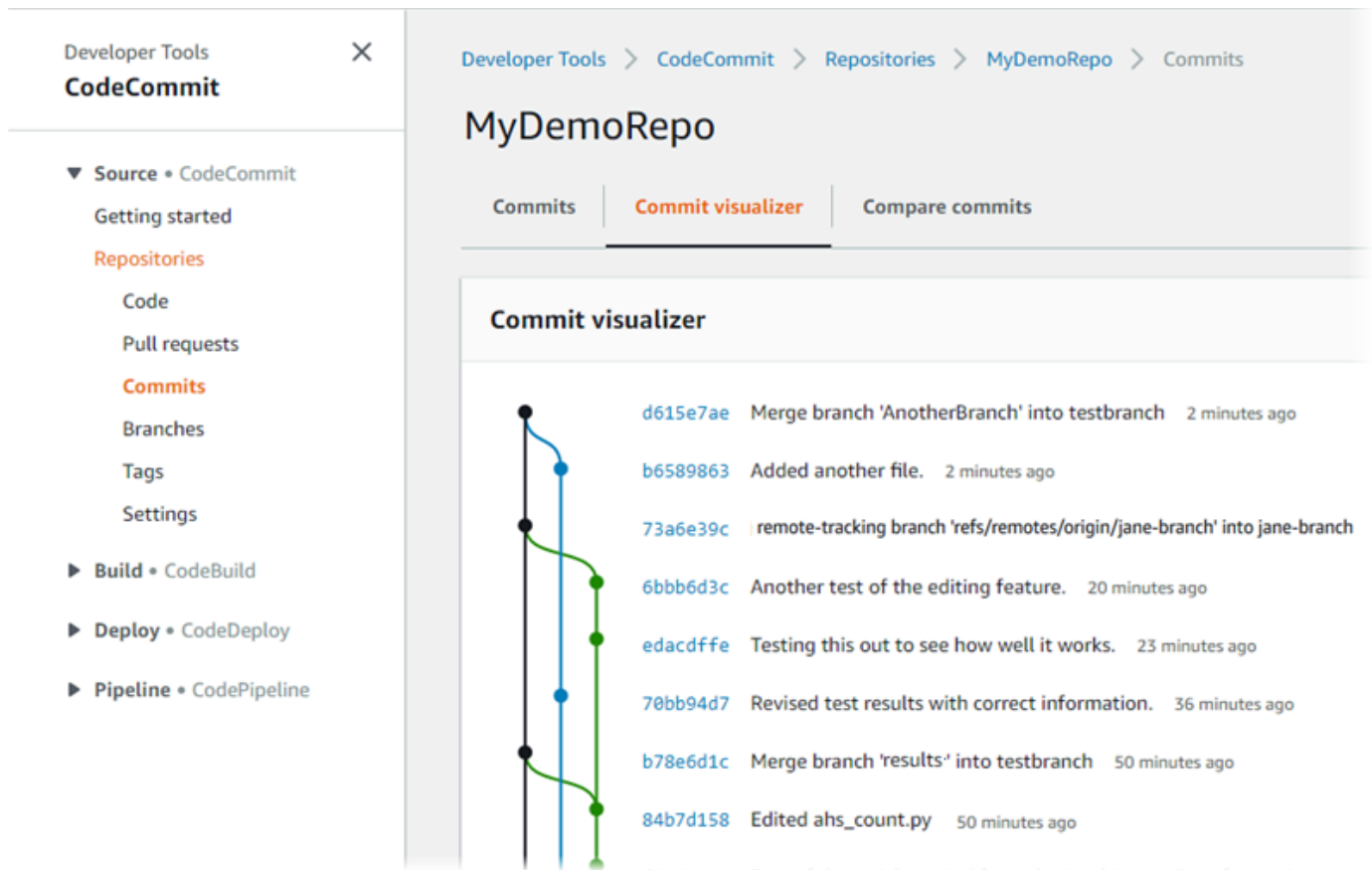
Para obtener más información, consulte [Comentar una confirmación](#).

8. Para ver las diferencias entre dos especificadores de confirmaciones, incluidas las etiquetas, las ramas y las confirmaciones, en el panel de navegación IDs, selecciona Confirmaciones y, a continuación, selecciona Comparar confirmaciones.

The screenshot shows the 'Compare commits' page in the AWS CodeCommit console for a repository named 'MyDemoRepo'. The breadcrumb navigation at the top reads: 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Compare'. Below the repository name, there are three tabs: 'Commits', 'Commit visualizer', and 'Compare commits' (which is selected and highlighted in orange). The 'Compare commits' section includes a 'Destination' dropdown set to 'AnotherBranch' and a 'Source' dropdown set to '6b65eb76'. There are 'Compare' and 'Cancel' buttons. Below this, there's a navigation bar with '< Page 1 of 1 >', a 'Go to file' search box, and toggle options for 'Hide whitespace changes' (checked), 'Unified' (selected), and 'Split'. The main content area displays a diff for the file 'ahs_count.py'. It shows a comparison between two versions of the file, with line numbers 5 through 10. Line 8 has a red background, indicating a deletion in the source commit, and line 9 has a green background, indicating an addition in the destination commit. The diff shows changes to the 'ahs' variable and the 'print' statements. Below the diff, there's a section for 'anothernew/dir2/anothertest.txt' marked as 'Added', with 'Browse file contents' and 'Comment on file' buttons.

Para obtener más información, consulte [Examinación del historial de confirmaciones de un repositorio](#) y [Comparar confirmaciones](#).

9. En Commits (Confirmaciones), seleccione la pestaña Commit visualizer (Visualizador de confirmaciones).



Aparece el gráfico de la confirmación, que muestra la línea de asunto de cada confirmación junto a su punto en el gráfico. La longitud de la línea de asunto está limitada a 80 caracteres.

- Para ver más información acerca de una confirmación, elija el ID de confirmación abreviado. Para representar el grafo de una confirmación específica, elija ese punto en el grado. Para obtener más información, consulte [Visualización de un gráfico del historial de confirmaciones de un repositorio](#).

Paso 4: Crear y colaborar en una solicitud de extracción

Cuando se trabaja en un repositorio con otros usuarios, es posible colaborar en el código y revisar los cambios. Puede crear una solicitud de extracción para que otros usuarios revisen y comenten los cambios de código en una ramificación. También puede crear una o varias reglas de aprobación para la solicitud de extracción. Por ejemplo, puede crear una regla de aprobación que requiera que al menos otros dos usuarios aprueben la solicitud de extracción para que se pueda combinar. Una vez aprobada la solicitud de extracción, puede combinar esos cambios en su ramificación de destino. Si configura las notificaciones en el repositorio, los usuarios del repositorio podrán recibir mensajes de correo electrónico acerca de eventos del repositorio (por ejemplo, para solicitudes de extracción).

o cuando alguien añade comentarios sobre el código). Para obtener más información, consulte [Configurar las notificaciones para los eventos en una AWS CodeCommit repositorio](#).

 Important

Antes de crear una solicitud de extracción, debe crear una ramificación que contenga los cambios de código que desea revisar. Para obtener más información, consulte [Creación de una ramificación](#).

Para crear y colaborar en una solicitud de extracción

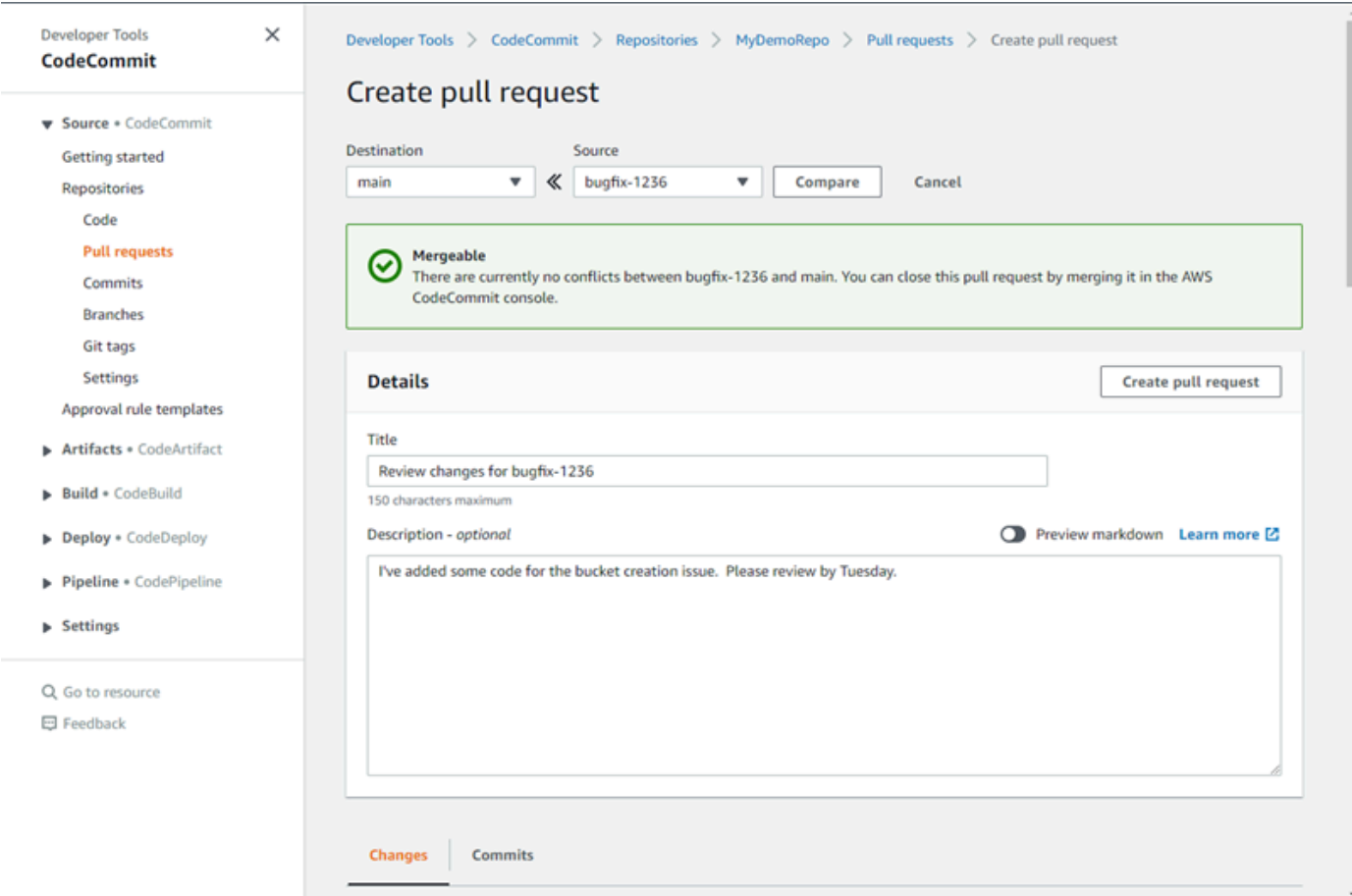
1. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
2. En Pull request (Solicitud de extracción), consulte Create pull request (Crear solicitud de extracción).

 Tip

También puede crear solicitudes de extracción en Branches (Ramificaciones) y Code (Código).

En Create pull request, en Source, seleccione la ramificación que contiene los cambios que desea revisar. En Destination (Destino), seleccione la ramificación en la que desea revisar el código que se va a combinar cuando se cierre la solicitud de extracción. Elija Compare.

3. Revise los detalles de la combinación y los cambios para confirmar que la solicitud de extracción contiene los cambios y las confirmaciones que desea revisar. De ser así, en Title (Título), introduzca un título para la revisión. Este es el título que aparece en la lista de solicitudes de extracción para el repositorio. En Description, introduzca información sobre de qué trata esta revisión y cualquier otro dato que pueda ser de utilidad a los revisores. Seleccione Crear.



4. Su solicitud de extracción aparece en la lista de las solicitudes de extracción del repositorio. Puede filtrar la vista para mostrar solo las solicitudes abiertas, las solicitudes cerradas, las solicitudes que ha creado y mucho más.

Pull requests Info					
Open pull requests Create pull request					
Pull request	Author	Destination	Last activity	Status	Approval status
31: testing this	Saanvi_Sarkar	preprod	4 minutes ago	Open	No approval rules
25: Updated some of our Java samples	Li_Juan	main	5 minutes ago	Open	0 of 1 rules satisfied
29: Changing duplicate value	Li_Juan	main	19 minutes ago	Open	0 of 1 rules satisfied
22: Test pull request	Saanvi_Sarkar	preprod	20 minutes ago	Open	No approval rules
28: Changes to some of our code samples	Li_Juan	main	1 month ago	Open	0 of 1 rules satisfied
20: A bugfix to add missing examples for S3	Saanvi_Sarkar	main	1 month ago	Open	0 of 1 rules satisfied

5. Puede añadir una regla de aprobación a la solicitud de extracción para asegurarse de que se cumplen determinadas condiciones antes de que se pueda combinar. Para agregar una regla de aprobación a la solicitud de extracción, seleccione la solicitud de extracción de la lista. En la pestaña Approvals (Aprobaciones), elija Create approval rule (Crear regla de aprobación).

6. En Rule name (Nombre de regla), indique un nombre descriptivo para la regla. Por ejemplo, si desea requerir que dos personas aprueben una solicitud de extracción para poder combinarla, puede asignar a la regla el nombre **Require two approvals before merge**. En Number of approvals needed (Número de aprobaciones necesarias), escriba **2**, que es el número deseado. El valor predeterminado es 1. Seleccione Enviar. Para obtener más información sobre las reglas de aprobación y los miembros del grupo de aprobación, consulte [Crear una regla de aprobación para una solicitud de extracción](#).

Create approval rule

Rule details

Rule name
Require two approvals before merge

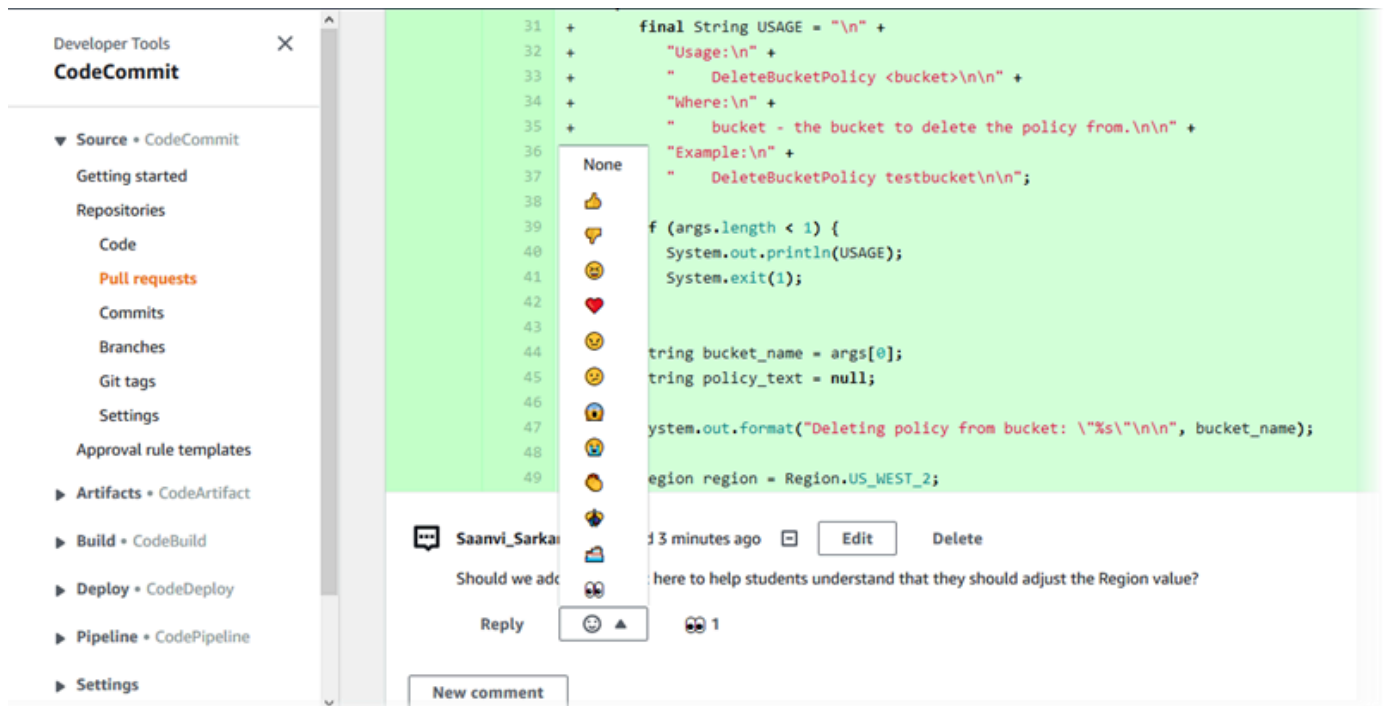
Number of approvals needed
2

Approval pool members - *optional*
If approval pool members are specified, only approvals from these members will count toward satisfying this rule. Use a wildcard to match multiple approvers with one value.

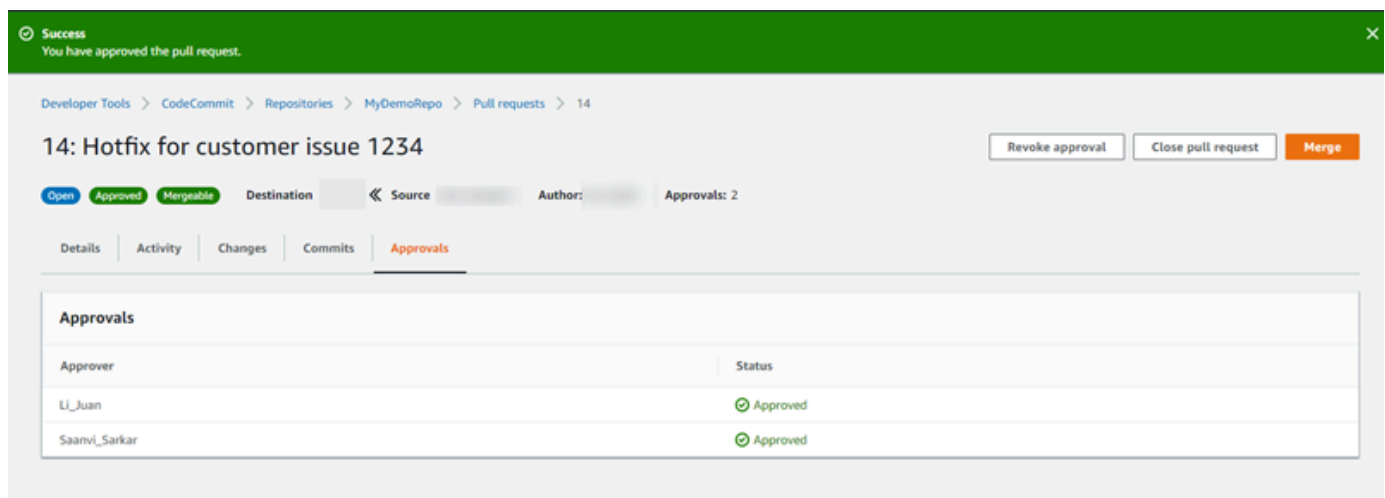
Add

Cancel Submit

7. Si ha configurado notificaciones para su repositorio y ha elegido notificar a los usuarios acerca de los eventos de solicitud de extracción, los usuarios recibirán un correo electrónico acerca de la nueva solicitud de extracción. Los usuarios pueden ver los cambios y hacer comentarios acerca de determinadas líneas de código, archivos específicos y la propia solicitud de extracción. También pueden responder a los comentarios con texto y emojis. Si fuese necesario, puede insertar cambios en la ramificación de la solicitud de extracción, lo que actualizará la solicitud de extracción.



8. Si está satisfecho con los cambios realizados en la solicitud, elija Approve (Aprobar). Puede optar por aprobar una solicitud de extracción aunque no se hayan configurado reglas de aprobación para ella. Esto proporciona un registro claro de que ha revisado la solicitud de extracción y de que ha aprobado los cambios. También puede optar por revocar su aprobación si cambia de opinión.



Note

No puede aprobar una solicitud de extracción si es usted quien la ha creado.

9. Cuando considere que se han revisado todos los cambios en el código y que se está de acuerdo con estos cambios, en la solicitud de extracción, realice una de las siguientes acciones:
 - Si desea cerrar la solicitud de extracción sin combinar las ramificaciones, elija Close pull request (Cerrar solicitud de extracción).
 - Si desea combinar las ramificaciones y cerrar la solicitud de extracción, elija Merge (Combinar). Puede elegir entre las estrategias de combinación disponibles para su código, que dependen de las diferencias entre la ramificación de origen y la de destino, y si se elimina automáticamente la ramificación de origen después de que la fusión se haya completado. Después que haya elegido sus opciones, elija Merge pull request (Combinar solicitud de extracción) para completar la combinación.

Merge pull request 9: Bug fix for unhandled exception

Merge request details

Pull request: #9 Bug fix for unhandled exception

Destination main  **Source** bugfix-bug1234

Merge strategy [Info](#)

Determines the way in which the current pull request will be merged into the destination branch

☐ Fast forward merge

`git merge --ff-only`

Merges the branches and moves the destination branch pointer to the tip of the source branch. This is the default merge strategy in Git.



☒ Squash and merge

`git merge --squash`

Combine all commits from the source branch into a single merge commit in the destination branch.



☐ 3-way merge

`git merge --no-ff`

Create a merge commit and adds individual source commits to the destination branch.



Commit message - *optional*

☐ Preview markdown

Squashed commit of the following

commit d49940ad

Author: Li Juan <li_juan@example.com>

Date: Tue May 07 2019 15:12:48 GMT-0700 (Pacific Daylight Time)

Fixing the bug reported in 1234.

Author name

Maria Garcia

Email address

maria_garcia@example.com

☒ Delete source branch bugfix-bug1234 after merging?

Cancel

Merge pull request

- Si hay conflictos de fusión en las ramas que no se pueden resolver automáticamente, puedes resolverlos en la CodeCommit consola o puedes usar tu cliente Git local para fusionar las ramas y, a continuación, ejecutar la fusión. Para obtener más información, consulte [Resolver conflictos en una solicitud de extracción en un repositorio de AWS CodeCommit](#).

 Note

Siempre puede combinar las ramificaciones manualmente, incluidas las ramificaciones de solicitud de extracción, mediante el comando `git merge` del repositorio local e introduciendo los cambios que desee efectuar.

Para obtener más información, consulte [Trabajo con solicitudes de extracción](#) y [Trabajar con plantillas de reglas de aprobación](#).

Paso 5: Eliminar

Si ya no necesitas el CodeCommit repositorio, debes eliminar el CodeCommit repositorio y otros recursos que hayas utilizado en este ejercicio para que no se te siga cobrando por el espacio de almacenamiento.

 Important

Esta acción no se puede deshacer. Una vez que haya eliminado este repositorio, no podrá clonarlo en ningún repositorio local o repositorio compartido. Tampoco podrá extraer datos de él, enviarle datos o realizar operaciones de Git desde cualquier repositorio local o repositorio compartido.

Si has configurado las notificaciones para tu repositorio, al eliminar el repositorio también se elimina la regla de Amazon CloudWatch Events creada para el repositorio. No se elimina el tema de Amazon SNS utilizado como destino de dicha regla.

Si ha configurado uno o más disparadores en el repositorio, al eliminar el repositorio no se eliminarán los temas de Amazon SNS ni las funciones de Lambda que configuró como objetivos de dichos disparadores. Asegúrese de eliminar esos recursos si no los necesita.

Para obtener más información, consulte [Eliminación de disparadores de un repositorio](#).

Para eliminar el repositorio CodeCommit

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. En Repositories (Repositorios), seleccione el repositorio que desee eliminar. Si ha seguido la convención de nomenclatura que se indica en este tema, este repositorio se llama MyDemoRepo.
3. En el panel de navegación, seleccione Configuración.
4. En la página Settings, en Delete repository, elija Delete repository.
5. Escriba **delete** y seleccione Delete (Eliminar). El repositorio se eliminará permanentemente.

Paso 6: Sigüientes pasos

Ahora que se ha familiarizado con CodeCommit algunas de sus funciones, considere la posibilidad de hacer lo siguiente:

- Si eres nuevo en Git CodeCommit o quieres revisar ejemplos del uso de Git con CodeCommit, continúa con el [Primeros pasos con Git y CodeCommit](#) tutorial.
- Si quieres trabajar con otras personas en un CodeCommit repositorio, consulta [Compartir un repositorio](#). (Si desea compartir el repositorio con usuarios de otra cuenta de Amazon Web Services, consulte [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#)).
- Si quieres migrar un repositorio a CodeCommit, sigue los pasos que se indican en [Migrar a CodeCommit](#).
- Si desea añadir el repositorio a una canalización de entrega continua, siga los pasos que se detallan en [Tutorial: crear una canalización](#).
- Si desea obtener más información sobre los productos y servicios que se integran con CodeCommit, así como ejemplos de la comunidad, consulte [Integraciones de productos y servicios](#).

Primeros pasos con Git y AWS CodeCommit

Si eres nuevo en Git CodeCommit, este tutorial te ayudará a aprender algunos comandos sencillos para empezar. Si ya está familiarizado con Git, puede omitir este tutorial y pasar al [Empezando con CodeCommit](#).

En este tutorial, crearás un repositorio que represente una copia local del CodeCommit repositorio, al que nos referimos como repositorio local.

Después de crear el repositorio local, deberá realizar algunos cambios en él. A continuación, envías (envías) los cambios al CodeCommit repositorio.

También simulas un entorno de equipo en el que dos usuarios envían los cambios de forma independiente a su repositorio local y los envían al CodeCommit repositorio. A continuación, los usuarios extraen los cambios del CodeCommit repositorio a su propio repositorio local para ver los cambios realizados por el otro usuario.

También puedes crear ramas y etiquetas y administrar algunos permisos de acceso al CodeCommit repositorio.

Cuando termine este tutorial, tendrá conocimientos suficientes de los conceptos principales de Git y CodeCommit y podrá utilizarlos en sus propios proyectos.

Realice los [requisitos previos y la configuración](#), lo que incluye:

- Asignar permisos al usuario de IAM.
- Configúrelo CodeCommit para conectarse a un repositorio mediante [HTTPS](#), SSH o [git-remote-codecommit](#). Para obtener más información sobre estas opciones, consulte [Configuración de AWS CodeCommit](#).
- Configure AWS CLI si desea utilizar la línea de comandos o el terminal para todas las operaciones, incluida la creación del repositorio.

Temas

- [Paso 1: Crear un CodeCommit repositorio](#)
- [Paso 2: Crear un repositorio local](#)
- [Paso 3: Crear la primera confirmación](#)
- [Paso 4: Enviar la primera confirmación](#)
- [Paso 5: Comparte el CodeCommit repositorio y envía y extrae otra confirmación](#)
- [Paso 6: Crear y compartir una ramificación](#)
- [Paso 7: Crear y compartir una etiqueta](#)
- [Paso 8: Configurar los permisos de acceso](#)
- [Paso 9: limpiar](#)

Paso 1: Crear un CodeCommit repositorio

En este paso, utilizará la CodeCommit consola para crear el repositorio.

Puede omitir este paso si ya tiene un CodeCommit repositorio que desee usar.

Note

Según el uso, es posible que se le cobre por crear un repositorio o por acceder a él. Para obtener más información, consulta [los precios](#) en la página de información CodeCommit del producto.

Para crear el CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Usa el selector de regiones para elegir Región de AWS dónde quieres crear el repositorio. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositorios, seleccione Crear repositorio.
4. En la página Create repository (Crear repositorio), en Repository name (Nombre del repositorio), escriba un nombre para el repositorio (por ejemplo, **MyDemoRepo**).

Note

En los nombres de los repositorios se distinguen mayúsculas de minúsculas y no pueden tener más de 100 caracteres. Para obtener más información, consulte [Límites](#).

5. (Opcional) En Description (Descripción), introduzca una descripción (por ejemplo, **My demonstration repository**). Esto puede ayudarle a usted y a otros usuarios a conocer el propósito del repositorio.
6. (Opcional) Seleccione Añadir etiqueta para añadir una o más etiquetas de repositorio (una etiqueta de atributo personalizada que le ayuda a organizar y gestionar sus AWS recursos) a su repositorio. Para obtener más información, consulte [Etiquetar repositorios en AWS CodeCommit](#).
7. (Opcional) Amplíe la configuración adicional para especificar si desea utilizar la clave predeterminada Clave administrada de AWS o la propia clave gestionada por el cliente para cifrar y descifrar los datos de este repositorio. Si decide utilizar su propia clave gestionada por el cliente, debe asegurarse de que esté disponible en el Región de AWS lugar donde vaya a crear

el repositorio y de que la clave esté activa. Para obtener más información, consulte [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#).

8. (Opcional) Seleccione Habilitar Amazon CodeGuru Reviewer para Java y Python si este repositorio va a contener código de Java o Python y desea que CodeGuru Reviewer analice ese código. CodeGuru Reviewer utiliza varios modelos de aprendizaje automático para detectar defectos en el código y sugerir automáticamente mejoras y correcciones en las solicitudes de extracción de cambios. Para obtener más información, consulte la Guía del usuario de Amazon CodeGuru Reviewer.
9. Seleccione Crear.

Note

El resto de los pasos de este tutorial se utilizan MyDemoRepo para el nombre del CodeCommit repositorio. Si elige otro nombre, asegúrese de utilizarlo durante todo el tutorial.

Para obtener más información sobre la creación de repositorios, incluso cómo crear un repositorio desde el terminal o línea de comandos, consulte [Creación de un repositorio](#).

Paso 2: Crear un repositorio local

En este paso, debe configurar un repositorio local en el equipo local para que se conecte a su repositorio. Para ello, seleccione un directorio en su equipo local que represente al repositorio local. Usas Git para clonar e inicializar una copia de tu CodeCommit repositorio vacío dentro de ese directorio. A continuación, especifique el nombre de usuario de Git y la dirección de correo electrónico utilizada para registrar sus confirmaciones.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. Encuentre en la lista el repositorio con el que desea establecer la conexión y selecciónelo. Elija Clone URL (Clonar URL), y, a continuación, elija el protocolo que desea utilizar al clonar o conectarse al repositorio. Con esto se copia la URL clonada.
 - Copie la URL HTTPS si utiliza credenciales de Git con su usuario de IAM o el ayudante de credenciales incluido con AWS CLI.

- Copie la dirección URL HTTPS (GRC) si utiliza el comando `git-remote-codecommit` en su equipo local.
- Copie la URL de SSH si utiliza un public/private key pair de claves SSH con su usuario de IAM.

 Note

Si ves una página de bienvenida en lugar de una lista de repositorios, significa que no hay ningún repositorio asociado a tu AWS cuenta en la Región de AWS que hayas iniciado sesión. Para crear un repositorio, consulte [the section called “Creación de un repositorio”](#) o siga los pasos del tutorial [Primeros pasos con Git y CodeCommit](#).

4. (Opcional) Le recomendamos que configure su cliente Git local para que **main** lo use como nombre de la ramificación predeterminada de su repositorio. Este es el nombre que se usa para la ramificación predeterminada en todos los ejemplos de esta guía. También es el mismo nombre de sucursal predeterminado que se CodeCommit usa si realizas tu primera confirmación en la consola. Ejecute el siguiente comando para configurar el nombre de ramificación predeterminado de forma global para su sistema:

```
git config --global init.defaultBranch main
```

Si prefiere usar un nombre de ramificación predeterminado diferente para todos sus repositorios, sustituya **main** por el nombre que prefiera. En este tutorial se asume que su ramificación predeterminada se denomina principal.

Si quiere usar diferentes nombres de ramificación predeterminados para diferentes repositorios, puede configurar este atributo localmente (`--local`) en lugar de globalmente (`--global`).

5. En el terminal o símbolo del sistema, clone el repositorio con el comando `git clone` y proporcione la URL de clonación que copió en el paso 3. La URL de clonación depende del protocolo y la configuración que utilice. Por ejemplo, si utilizas HTTPS con credenciales de Git para clonar un repositorio denominado *MyDemoRepo* en la región EE.UU. Este (Ohio):

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Si utiliza HTTPS con `git-remote-codecommit`:

```
git clone codecommit://MyDemoRepo my-demo-repo
```

Si utiliza SSH:

```
git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

 Note

Si aparece un error al intentar clonar un repositorio, es posible que no haya completado la configuración necesaria para el equipo local. Para obtener más información, consulte [Configuración de AWS CodeCommit](#).

Paso 3: Crear la primera confirmación

En este paso, debe crear la primera confirmación en su repositorio local. Para ello, cree dos archivos de ejemplo en su repositorio local. Puede utilizar Git para preparar el cambio y, a continuación, confirmarlo en el repositorio local.

1. Utilice un editor de texto para crear los siguientes dos archivos de texto de ejemplo en su directorio. Asigne un nombre a los archivos `cat.txt` y `dog.txt`:

```
cat.txt
-----
The domestic cat (Felis catus or Felis silvestris catus) is a small, usually furry,
domesticated, and carnivorous mammal.
```

```
dog.txt
-----
The domestic dog (Canis lupus familiaris) is a canid that is known as man's best
friend.
```

2. Ejecuta `git config` este procedimiento para añadir tu nombre de usuario y dirección de correo electrónico, representados por marcadores de posición, *your-user-name* y *your-email-address* a tu repositorio local. Esto hace que sea más fácil identificar las confirmaciones que realice:

```
git config --local user.name "your-user-name"  
git config --local user.email your-email-address
```

3. Si no configuró su nombre de ramificación predeterminado de forma global cuando creó el repositorio local, ejecute el siguiente comando para establecer el nombre de ramificación predeterminado como **main**:

```
git config --local init.defaultBranch main
```

4. Ejecute git add para preparar el cambio:

```
git add cat.txt dog.txt
```

5. Ejecute git commit para confirmar el cambio:

```
git commit -m "Added cat.txt and dog.txt"
```

Tip

Para ver la información detallada acerca de la confirmación que acaba de realizar, ejecute git log.

Paso 4: Enviar la primera confirmación

En este paso, envías la confirmación de tu repositorio local a tu repositorio. CodeCommit

Ejecuta git push para enviar tu confirmación a través del nombre remoto predeterminado que Git usa para tu CodeCommit repositorio (**origin**), desde la rama predeterminada de tu repositorio local (**main**):

```
git push -u origin main
```

 Tip

Una vez que hayas subido los archivos a tu CodeCommit repositorio, puedes usar la CodeCommit consola para ver el contenido. Para obtener más información, consulte [Navegación por los archivos en un repositorio](#).

Paso 5: Comparte el CodeCommit repositorio y envía y extrae otra confirmación

En este paso, compartes información sobre el CodeCommit repositorio con otro miembro del equipo. El miembro del equipo usa esta información para obtener una copia local, realizar algunos cambios en ella y, a continuación, enviar la copia local modificada a tu CodeCommit repositorio. A continuación, extraes los cambios del CodeCommit repositorio a tu repositorio local.

En este tutorial, debe simular el usuario de su colega haciendo que Git cree un directorio distinto al que ha creado en el [paso 2](#). (Normalmente, este directorio está en otro equipo). Este nuevo directorio es una copia de tu CodeCommit repositorio. Los cambios que realice en el directorio existente o en este nuevo directorio se realizan de manera independiente. La única forma de identificar los cambios en estos directorios es extraerlos del CodeCommit repositorio.

Aunque están en el mismo equipo local, denominamos al directorio existente su repositorio local y al nuevo directorio, el repositorio compartido.

Desde el nuevo directorio, obtendrá una copia independiente del CodeCommit repositorio. A continuación, añades un nuevo archivo de ejemplo, confirmas los cambios en el repositorio compartido y, a continuación, transfieres la confirmación del repositorio compartido a tu CodeCommit repositorio.

Por último, extraiga los cambios de su repositorio a su repositorio local y, a continuación, examínelo para ver los cambios que ha confirmado el otro usuario.

1. Cambie al directorio /tmp o al directorio c:\temp.
2. Ejecute git clone para extraer una copia del repositorio en el repositorio compartido:

Para HTTPS:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
shared-demo-repo
```

Para HTTPS con git-remote-codecommit:

```
git clone codecommit://MyDemoRepo shared-demo-repo
```

Para SSH:

```
git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo shared-demo-repo
```

 Note

Al clonar un repositorio mediante SSH en los sistemas operativos Windows, debe agregar el ID de la clave de SSH a la cadena de conexión de la siguiente manera:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Para obtener más información, consulte [Para conexiones SSH en Windows](#).

En este comando, MyDemoRepo está el nombre de tu repositorio. CodeCommit shared-demo-repo es el nombre del directorio que Git crea en el /tmp directorio o en el c:\temp directorio. Después de crear el directorio, Git extrae una copia de su repositorio en el directorio shared-demo-repo.

3. Cambie al directorio shared-demo-repo:

```
(For Linux, macOS, or Unix) cd /tmp/shared-demo-repo  
(For Windows) cd c:\temp\shared-demo-repo
```

4. Ejecute git config para añadir otro nombre de usuario y dirección de correo electrónico representados por marcadores de posición *other-user-name* y *other-email-address*. De ese modo, resulta más sencillo identificar las confirmaciones que ha realizado el otro usuario:

```
git config --local user.name "other-user-name"  
git config --local user.email other-email-address
```

- Utilice un editor de texto para crear el siguiente archivo de texto de ejemplo en el directorio `shared-demo-repo`. Asigne el nombre `horse.txt` al archivo:

```
horse.txt
-----
The horse (Equus ferus caballus) is one of two extant subspecies of Equus ferus.
```

- Ejecute `git add` para preparar el cambio en el repositorio compartido:

```
git add horse.txt
```

- Ejecute `git commit` para confirmar el cambio en el repositorio compartido:

```
git commit -m "Added horse.txt"
```

- Ejecuta `git push` para enviar tu confirmación inicial al nombre remoto predeterminado que Git usa para tu CodeCommit repositorio (`origin`), desde la rama predeterminada de tu repositorio local (`main`):

```
git push -u origin main
```

- Cambia a tu repositorio local y ejecuta `git pull` para acceder a tu repositorio local la confirmación que el repositorio compartido hizo en el repositorio. CodeCommit A continuación, ejecute `git log` para ver la confirmación que se inició desde el repositorio compartido.

Paso 6: Crear y compartir una ramificación

En este paso, creas una rama en tu repositorio local, realizas algunos cambios y, a continuación, insertas la rama en tu repositorio. CodeCommit A continuación, llevas la rama al repositorio compartido desde tu CodeCommit repositorio.

Una ramificación permite desarrollar de forma independiente una versión diferente del contenido del repositorio (por ejemplo, trabajar en una nueva característica de software sin que afecte al trabajo de los otros miembros del equipo). Cuando esta característica se estabilice, deberá combinar la ramificación con una ramificación más estable del software.

Utilice Git para crear la ramificación y, a continuación, dirigirla a la primera confirmación que ha realizado. Usas Git para empujar la rama al CodeCommit repositorio. A continuación, cambie al

repositorio compartido y use Git para extraer la nueva ramificación en su repositorio local compartido y examinar la ramificación.

1. Desde su repositorio local, ejecute `git checkout` y especifique el nombre de la ramificación (por ejemplo, `MyNewBranch`) y el ID de la primera confirmación realizada en el repositorio local.

Si desconoce el ID de confirmación, ejecute `git log` para obtenerlo. Asegúrese de que la confirmación tiene su nombre de usuario y dirección de correo electrónico, y no el nombre de usuario y la dirección de correo electrónico del otro usuario. Esto es para simular que `main` se trata de una versión estable del CodeCommit repositorio y que la `MyNewBranch` rama sirve para alguna función nueva y relativamente inestable:

```
git checkout -b MyNewBranch commit-ID
```

2. Ejecuta `git push` para enviar la nueva rama desde el repositorio local al CodeCommit repositorio:

```
git push origin MyNewBranch
```

3. Ahora, extraiga la ramificación en el repositorio compartido y compruebe los resultados:
 1. Cambie al directorio de repositorio compartido (`shared-demo-repo`).
 2. Extraiga la nueva ramificación (`git fetch origin`).
 3. Confirme que la ramificación se ha extraído (`git branch --all` muestra una lista de todas las ramificaciones del repositorio).
 4. Cambie a la nueva ramificación (`git checkout MyNewBranch`).
 5. Confirme que ha cambiado a la ramificación `MyNewBranch` ejecutando `git status` o `git branch`. El resultado muestra en qué ramificación se encuentra. En este caso debería ser `MyNewBranch`.
 6. Vea la lista de confirmaciones en la ramificación (`git log`).

Aquí tiene la lista de comandos Git que puede llamar:

```
git fetch origin
git branch --all
git checkout MyNewBranch
git branch or git status
git log
```

4. Vuelva a la ramificación `main` y consulte la lista de confirmaciones. Los comandos Git deberían tener el siguiente aspecto:

```
git checkout main
git log
```

5. Cambie a la ramificación `main` de su repositorio local. Puede ejecutar `git status` o `git branch`. El resultado muestra en qué ramificación se encuentra. En este caso debería ser `main`. Los comandos Git deberían tener el siguiente aspecto:

```
git checkout main
git branch or git status
```

Paso 7: Crear y compartir una etiqueta

En este paso, creas dos etiquetas en tu repositorio local, las asocias a las confirmaciones y, a continuación, las insertas en tu CodeCommit repositorio. A continuación, extraes los cambios del CodeCommit repositorio al repositorio compartido.

Las etiquetas se utilizan para asignar un nombre legible para los humanos a una confirmación (o ramificación o incluso a otra etiqueta). Las puede utilizar, por ejemplo, si desea etiquetar una confirmación como `v2.1`. Una confirmación, ramificación o etiqueta puede tener un número indefinido de etiquetas asociado, pero una etiqueta individual solo puede asociarse a una única confirmación, ramificación o etiqueta. En este tutorial, etiquetará una confirmación como `release` y otra como `beta`.

Usará Git para crear las etiquetas. La etiqueta `release` apuntará a la primera confirmación que haya realizado y la etiqueta `beta`, a la confirmación que haya realizado otro usuario. A continuación, utilizas Git para enviar las etiquetas al CodeCommit repositorio. A continuación, cambie al repositorio compartido y use Git para extraer las etiquetas en su repositorio local compartido y examínelas.

1. Desde su repositorio local, ejecute `git tag` y especifique el nombre de la nueva etiqueta (`release`) y el ID de la primera confirmación realizada en el repositorio local.

Si desconoce el ID de confirmación, ejecute `git log` para obtenerlo. Asegúrese de que la confirmación tiene su nombre de usuario y dirección de correo electrónico, y no el nombre

de usuario y la dirección de correo electrónico del otro usuario. Esto es para simular que tu confirmación es una versión estable del CodeCommit repositorio:

```
git tag release commit-ID
```

Ejecute git tag de nuevo para etiquetar la confirmación del otro usuario con la etiqueta beta. Estos pasos se realizan para simular que la confirmación es para una característica nueva relativamente inestable:

```
git tag beta commit-ID
```

2. Ejecuta git push --tags para enviar las etiquetas al CodeCommit repositorio.
3. Ahora, extraiga las etiquetas en el repositorio compartido y compruebe los resultados:
 1. Cambie al directorio de repositorio compartido (shared-demo-repo).
 2. Extraiga las nuevas etiquetas (git fetch origin).
 3. Confirme que las etiquetas se han extraído (git tag muestra una lista de etiquetas para el repositorio).
 4. Consulte la información sobre cada etiqueta (git log release y git log beta).

Aquí tiene la lista de comandos Git que puede llamar:

```
git fetch origin  
git tag  
git log release  
git log beta
```

4. Pruébalo también en el repositorio local:

```
git log release  
git log beta
```

Paso 8: Configurar los permisos de acceso

En este paso, le das permiso a un usuario para sincronizar el repositorio compartido con el CodeCommit repositorio. Se trata de un paso opcional. Se recomienda a los usuarios que estén

interesados en aprender cómo controlar el acceso a los CodeCommit repositorios cuando los usuarios utilizan credenciales de Git o se utilizan pares de claves SSH con los usuarios de IAM para acceder a los repositorios. CodeCommit

 Note

Si utiliza un acceso federado, credenciales temporales o un proveedor de identidad web, como el IAM Identity Center, configure los usuarios, el acceso y los permisos de su proveedor de identidades y, a continuación, use git-remote-codecommit. Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#) y [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#).

Para ello, utilice la consola de IAM para crear un usuario que, de forma predeterminada, no tenga permisos para sincronizar el repositorio compartido con el repositorio. CodeCommit Puede ejecutar git pull para comprobarlo. Si el nuevo usuario no tiene permiso para sincronizar, el comando no funciona. A continuación, vuelva a la consola de IAM y aplique una política que permita al usuario utilizar git pull. Vuelva a ejecutar git pull para comprobarlo.

Este paso se ha redactado desde el supuesto de que tiene permisos para crear usuarios de IAM en su cuenta de Amazon Web Services. Si no dispone de estos permisos, no podrá realizar los procedimientos de este paso. Vaya a [Paso 9: limpiar](#) para eliminar los recursos que ha utilizado para el tutorial.

1. Inicie sesión en la consola de Consola de administración de AWS IAM y ábrala en. <https://console.aws.amazon.com/iam/>

Asegúrese de iniciar sesión con el mismo nombre de usuario y contraseña que utilizó en [Configuración](#).

2. En el panel de navegación, seleccione Users y después Create New Users.
3. En el primer cuadro Enter User Names (Especificar nombres de usuario), introduzca un nombre de usuario de ejemplo (por ejemplo, **JaneDoe-CodeCommit**). Seleccione la casilla Generate an access key for each user (Generar una clave de acceso para cada usuario) y, a continuación, haga clic en Create (Crear).
4. Elija Show User Security Credentials. Anote el ID de clave de acceso y la clave de acceso secreta o elija Download Credentials.

5. Siga las instrucciones de [Para usuarios HTTPS mediante credenciales de Git](#) para generar y proporcionar las credenciales del usuario de IAM.

Si desea utilizar SSH, siga las instrucciones en [SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit](#) o [Paso 3: Configura las claves públicas y privadas para Git y CodeCommit](#) para configurar el usuario con claves públicas y privadas.

6. Ejecute `git pull`. Debería aparecer el siguiente error:

Para HTTPS:

```
fatal: unable to access 'https://git-codecommit.us-east-2.amazonaws.com/v1/repos/repository-name': The requested URL returned error: 403.
```

Para SSH:

```
fatal: unable to access 'ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/repository-name': The requested URL returned error: 403.
```

El error aparece porque el nuevo usuario no tiene permiso para sincronizar el repositorio compartido con el repositorio. CodeCommit

7. Vuelva a la consola de IAM. En el panel de navegación, seleccione Políticas y, a continuación, seleccione Create Policy. Si aparece el botón Empezar, selecciónelo y, a continuación, seleccione Crear política.
8. Junto a Create Your Own Policy, seleccione Select.
9. En el cuadro Policy Name (Nombre de política), introduzca un nombre (por ejemplo, **CodeCommitAccess-GettingStarted**).
10. En el cuadro Documento de política, introduzca lo siguiente, que permite a un usuario de IAM extraer contenido de cualquier repositorio asociado con el usuario de IAM:

JSON

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

```
"Action": [  
  "codecommit:GitPull"  
],  
"Resource": "*" ]  
}
```

 Tip

Si, por el contrario, desea que el usuario de IAM pueda enviar confirmaciones a cualquier repositorio asociado con el usuario de IAM, introduzca lo siguiente:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "codecommit:GitPull",  
        "codecommit:GitPush"  
      ],  
      "Resource": "*" ]  
    }  
  ]  
}
```

Para obtener información sobre otros permisos de CodeCommit acción y recursos que puede conceder a los usuarios, consulte. [Autenticación y control de acceso para AWS CodeCommit](#)

11. En el panel de navegación, seleccione Usuarios.
12. Elija el nombre de usuario de ejemplo (por ejemplo, **JaneDoe-CodeCommit**) al que desea asociar la política.
13. Elija la pestaña Permisos.

14. En Managed Policies, elija Attach Policy.
15. Seleccione la política **CodeCommitAccess-GettingStarted** que acaba de crear y, a continuación, elija Attach Policy (Asociar política).
16. Ejecute git pull. Esta vez el comando debería funcionar y debería aparecer un mensaje Already up-to-date.
17. Si utiliza HTTPS, cambie a sus credenciales Git originales o, si usa git-remote-codecommit, a su perfil habitual. Para obtener más información, consulte las instrucciones en [Configuración de usuarios HTTPS mediante credenciales de Gito](#) [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Si utiliza SSH, cambie a sus claves originales. Para obtener más información, consulte [SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit](#) o [Paso 3: Configura las claves públicas y privadas para Git y CodeCommit](#).

Ha llegado al final del tutorial.

Paso 9: limpiar

En este paso, eliminas el CodeCommit repositorio que utilizaste en este tutorial para que no se te siga cobrando por el espacio de almacenamiento.

También eliminas el repositorio local y el repositorio compartido de tu máquina local, ya que no serán necesarios después de eliminar el CodeCommit repositorio.

Important

Después de eliminar este repositorio, ya no podrá clonarlo en ningún repositorio local ni en ningún repositorio compartido. Tampoco podrá extraer ni enviar datos desde ningún repositorio local ni repositorio compartido. Esta acción no se puede deshacer.

Para eliminar el CodeCommit repositorio (consola)

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En la página Dashboard (Panel), en la lista de repositorios, elija MyDemoRepo.
3. En el panel de navegación, seleccione Configuración.
4. En la página Settings, en Delete repository, elija Delete repository.

5. En el cuadro junto a Type the name of the repository to confirm deletion (Escriba el nombre del repositorio para confirmar la eliminación), introduzca **MyDemoRepo**, a continuación, seleccione Delete (Eliminar).

Para eliminar el repositorio () CodeCommit AWS CLI

Ejecute el comando [delete-repository](#):

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Para eliminar el repositorio local y el repositorio compartido

Para Linux, macOS o Unix:

```
cd /tmp
rm -rf /tmp/my-demo-repo
rm -rf /tmp/shared-demo-repo
```

Para Windows:

```
cd c:\temp
rd /s /q c:\temp\my-demo-repo
rd /s /q c:\temp\shared-demo-repo
```

Integraciones de productos y servicios con AWS CodeCommit

De forma predeterminada, CodeCommit se integra con una serie de servicios de AWS. CodeCommit también se puede utilizar con productos y servicios que no son de AWS. La siguiente información puede ayudarle a configurar CodeCommit para integrarlo con los productos y servicios que utilice.

Temas

- [Integración con otros servicios de AWS](#)
- [Ejemplos de integración de la comunidad](#)

Integración con otros servicios de AWS

CodeCommit está integrado con los siguientes servicios de AWS:

AWS Amplify

[AWS Amplify](#) facilita crear, configurar e implementar aplicaciones móviles escalables con la tecnología de AWS. Amplify aprovisiona y administra de manera continua su backend móvil y ofrece un marco simple para integrar fácilmente su backend con sus frontends de iOS, Android, Web y React Native. Amplify también automatiza el proceso de liberación de la aplicación de frontend y backend lo que permite que ofrezca características más rápidas.

Puede conectar su repositorio de CodeCommit en la consola de Amplify. Después de autorizar la consola de Amplify, Amplify obtiene un token de acceso del proveedor del repositorio, pero no lo almacena en los servidores de AWS. Amplify obtiene acceso a su repositorio utilizando claves de implementación instaladas solo en un repositorio específico.

Más información:

- [AWS Amplify Guía del usuario de](#)
- [Introducción a](#)

AWS Cloud9

[AWS Cloud9](#) contiene una colección de herramientas que se utilizan para escribir código, así como para compilar, ejecutar, probar, depurar y publicar software en la nube. Este conjunto de herramientas se conoce como el entorno de desarrollo integrado o IDE de AWS Cloud9.

Es posible acceder al IDE de AWS Cloud9 a través de un navegador web. El IDE de ofrece una completa experiencia de edición de código, con soporte para varios lenguajes de programación y depuradores de tiempo de ejecución, además de un terminal integrado.

Más información:

- [AWS Cloud9 Guía del usuario de](#)
- [AWS CodeCommit Ejemplo de para AWS Cloud9](#)
- [Integre AWS Cloud9 con AWS CodeCommit](#)

CloudFormation

[CloudFormation](#) es un servicio que le ayuda a modelar y configurar sus recursos de AWS, por lo que podrá dedicar menos tiempo a la administración de dichos recursos y más tiempo a centrarse en sus aplicaciones. Puede crear una plantilla que describa los recursos, incluido un repositorio de CodeCommit. CloudFormation se encarga del aprovisionamiento y la configuración de dichos recursos.

Más información:

- [AWS CodeCommit Página de recursos del repositorio de](#)

AWS CloudTrail

[CloudTrail](#) captura las llamadas a las API de AWS y los eventos relacionados llevados a cabo por o en nombre de una cuenta de Amazon Web Services y envía archivos de registro al bucket de Amazon S3 que especifique. Puede configurar CloudTrail para que capture las llamadas a las API desde la consola AWS CodeCommit, los comandos de CodeCommit desde la AWS CLI, el cliente de Git local y desde la API de CodeCommit.

Más información:

- [Registrar llamadas a la AWS CodeCommit API con AWS CloudTrail](#)

Eventos de Amazon CloudWatch

[CloudWatch Events](#) proporciona un flujo de eventos de sistema casi en tiempo real que describe cambios en los recursos de AWS. Mediante reglas sencillas que puede configurar rápidamente, puede asignar los eventos y dirigirlos a uno o más flujos o funciones de destino. CloudWatch Events conoce los cambios operativos a medida que se producen. CloudWatch Events responde a estos cambios operativos y toma medidas según sea necesario, enviando mensajes para responder al entorno, activando funciones, realizando cambios y captando información de estado.

Puede configurar CloudWatch Events para que supervise los repositorios de CodeCommit y responda a los eventos del repositorio enviando flujos, funciones, tareas u otros procesos en otros servicios de AWS, como Amazon Simple Queue Service, Amazon Kinesis, AWS Lambda y muchos otros.

Más información:

- [Guía del usuario de CloudWatch Events](#)
- [AWS CodeCommit Eventos de](#)
- Entrada de blog: [Crear flujos de trabajo sin servidor de AWS CodeCommit usando los eventos de Amazon CloudWatch y JGit](#)

AWS CodeBuild

[CodeBuild](#) es un servicio de compilación completamente administrado en la nube que compila código fuente, ejecuta pruebas unitarias y produce artefactos listos para su implementación. Puede almacenar el código fuente que se compilará y las especificaciones de compilación en un repositorio de CodeCommit. Además, puede utilizar CodeBuild directamente con CodeCommit o puede incorporar CodeBuild y CodeCommit en una canalización de entrega continua con CodePipeline.

Más información:

- [Planificación de una compilación](#)
- [Crear un proyecto de compilación](#)
- [Uso de CodePipeline con AWS CodeBuild para ejecutar compilaciones](#)

Revisor de Amazon CodeGuru

El revisor de Amazon CodeGuru es un servicio automatizado de revisión de código que utiliza el análisis de programas y el machine learning para detectar problemas comunes y recomendar correcciones en el código Java o Python. Puede asociar los repositorios de la cuenta de Amazon Web Services al revisor de CodeGuru. Cuando lo hace, el revisor de CodeGuru crea un rol vinculado al servicio que permite al revisor de CodeGuru analizar el código en todas las solicitudes de extracción creadas después de realizar la asociación. Más información:

- [Amazon CodeGuru Reviewer y AWS CodeCommit](#)
- [Guía de usuario del revisor de Amazon CodeGuru](#)

AWS CodePipeline

[CodePipeline](#) es un servicio de entrega continua que puede utilizar para modelar, visualizar y automatizar los pasos necesarios para lanzar su software. Puede configurar CodePipeline para que utilice un repositorio de CodeCommit como una acción de origen en una canalización y automatizar la compilación, las pruebas y la implementación de sus cambios.

Más información:

- [Tutorial sencillo sobre canalizaciones con CodePipeline y AWS CodeCommit](#)
- [Migrar a la detección de cambios de los eventos de Amazon CloudWatch para las canalizaciones con un repositorio de CodeCommit](#)
- [Métodos de detección de cambios utilizados para iniciar canalizaciones automáticamente](#)

AWS Elastic Beanstalk

[Elastic Beanstalk](#) es un servicio administrado que facilita la implementación y la administración de las aplicaciones en la nube de AWS sin tener que preocuparse por la infraestructura que las ejecuta. Puede utilizar la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB) para implementar la aplicación directamente desde un repositorio de CodeCommit nuevo o existente.

Más información:

- [Uso de la CLI de EB con AWS CodeCommit](#)
- [Uso de un repositorio de AWS CodeCommit existente](#)
- [eb codesource \(comando de la CLI EB\)](#)

AWS Key Management Service

[AWS KMS](#) es un servicio administrado que le permite crear y controlar fácilmente las claves de cifrado que se utilizan para cifrar datos. De forma predeterminada, CodeCommit utiliza AWS KMS para cifrar repositorios.

Más información:

- [AWS KMS y cifrado](#)

AWS Lambda

[Lambda](#) le permite ejecutar código sin aprovisionar ni administrar servidores. Puede configurar disparadores para los repositorios de CodeCommit que invoca las funciones de Lambda en respuesta a los eventos del repositorio.

Más información:

- [Creación de un disparador para una función de Lambda](#)
- [AWS Lambda Guía para desarrolladores de](#)

Amazon Simple Notification Service

[Amazon SNS](#) es un servicio web que permite a las aplicaciones, los usuarios finales y los dispositivos enviar y recibir al instante notificaciones desde la nube. Puede configurar disparadores para los repositorios de CodeCommit que envía notificaciones de Amazon SNS en respuesta a los eventos del repositorio. También puede utilizar notificaciones de Amazon SNS para la integración con otros servicios de AWS. Por ejemplo, puede utilizar una notificación de Amazon SNS para enviar mensajes a una cola de Amazon Simple Queue Service.

Más información:

- [Creación de un disparador para un tema de Amazon SNS](#)
- [Guía para desarrolladores de Amazon Simple Notification Service](#)

Ejemplos de integración de la comunidad

En las siguientes secciones se incluyen enlaces a entradas de blogs, artículos y ejemplos proporcionados en la comunidad.

Note

Estos enlaces se ofrecen solo con fines informativos y no deben considerarse una lista completa o una anotación del contenido de los ejemplos. AWS no se hace responsable del contenido ni de la exactitud del contenido externo.

Temas

- [Publicaciones de blog](#)
- [Ejemplos de código](#)

Publicaciones de blog

- [Integración de SonarQube como aprobador de solicitudes de extracción en AWS CodeCommit](#)

Aprenda a crear un repositorio de CodeCommit que requiera un análisis de calidad de SonarQube satisfactorio antes de que se puedan combinar las solicitudes de extracción.

Fecha de publicación: 12 de diciembre de 2019

- [Migración a AWS CodeCommit, AWS CodePipeline y AWS CodeBuild desde GitLab](#)

Obtenga información sobre cómo migrar varios repositorios a AWS CodeCommit desde GitLab y configurar una canalización de CI/CD utilizando AWS CodePipeline y AWS CodeBuild.

Fecha de publicación: 22 de noviembre de 2019

- [Implementación de GitFlow usando AWS CodePipeline, AWS CodeCommit, AWS CodeBuild y AWS CodeDeploy](#)

Aprenda a implementar GitFlow usando AWS CodePipeline, AWS CodeCommit, AWS CodeBuild y AWS CodeDeploy.

Fecha de publicación: 22 de febrero de 2019

- [Uso de Git AWS CodeCommit en varias cuentas de AWS](#)

Aprenda a administrar su configuración de Git en varias cuentas de Amazon Web Services.

Fecha de publicación: 12 de febrero de 2019

- [Validación de solicitudes de extracción de AWS CodeCommit con AWS CodeBuild y AWS Lambda](#)

Aprenda a validar las solicitudes de extracción con AWS CodeCommit, AWS CodeBuild y AWS Lambda. Al ejecutar pruebas frente a los cambios propuestos antes de combinarlos en la ramificación predeterminada, puede ayudar a garantizar un alto nivel de calidad en las solicitudes de extracción, detectar cualquier problema potencial y aumentar la confianza del desarrollador en relación con sus cambios.

Fecha de publicación: 11 de febrero de 2019

- [Uso de identidades federadas con AWS CodeCommit](#)

Aprenda a acceder a repositorios en AWS CodeCommit usando las identidades utilizadas en su negocio.

Fecha de publicación: 5 de octubre de 2018

- [Perfeccionar el acceso a las ramificaciones en AWS CodeCommit](#)

Descubra cómo restringir las confirmaciones en las ramificaciones del repositorio creando y aplicando una política de IAM que utilice una clave de contexto.

Fecha de publicación: 16 de mayo de 2018

- [Replicar repositorios de AWS CodeCommit entre regiones mediante AWS Fargate](#)

Descubra cómo configurar una replicación continua de un repositorio de CodeCommit desde una región de AWS a otra mediante una arquitectura sin servidor.

Fecha de publicación: 11 de abril de 2018

- [Distribuir una infraestructura de AWS OpsWorks for Chef Automate](#)

Descubra cómo usar CodePipeline, CodeCommit, CodeBuild y AWS Lambda para garantizar que los libros de recetas y otras configuraciones se implementen de manera coherente en dos o más servidores de Chef que residan en una o varias Regiones de AWS.

Fecha de publicación: 9 de marzo de 2018

- [Peanut Butter y Chocolate: canalización CI/CD de funciones de Azure con AWS CodeCommit](#)

Descubra cómo crear una canalización CI/CD de funciones de Azure basada en PowerShell en la que el código se almacena en un repositorio de CodeCommit.

Fecha de publicación: 19 de febrero de 2018

- [Implementación continua en Kubernetes mediante AWS CodePipeline, AWS CodeCommit, AWS CodeBuild, Amazon ECR y AWS Lambda](#)

Descubra cómo usar Kubernetes y AWS para crear una canalización de implementación continua y completamente administrada para las aplicaciones basadas en contenedores.

Fecha de publicación: 11 de enero de 2018

- [Usar solicitudes de extracción de AWS CodeCommit para solicitar revisiones de código y comentar el código](#)

Descubra cómo usar las solicitudes de extracción para revisar, comentar e iterar de forma interactiva los cambios de código en un repositorio de CodeCommit.

Fecha de publicación: 20 de noviembre de 2017

- [Crear flujos de trabajo sin servidor de AWS CodeCommit usando los eventos de Amazon CloudWatch y JGit](#)

Aprenda a crear reglas de CloudWatch Events que procesen los cambios en un repositorio utilizando eventos del repositorio de CodeCommit y acciones de destino en otros servicios de AWS. Entre los ejemplos, se incluyen funciones de AWS Lambda que aplican las políticas de mensajes de confirmación de Git en confirmaciones, replican un repositorio de CodeCommit y realizan una copia de seguridad de un repositorio de CodeCommit en Amazon S3.

Fecha de publicación: 3 de agosto de 2017

- [Migración a AWS CodeCommit](#)

Aprenda a enviar código a dos repositorios como parte de la migración a la hora de utilizar otro repositorio Git en CodeCommit al usar SourceTree.

Fecha de publicación: 6 de septiembre de 2016

- [Configuración de pruebas continuas con Appium, AWS CodeCommit, Jenkins y AWS Device Farm](#)

Aprenda a crear un proceso de pruebas continuo para dispositivos móviles con Appium, CodeCommit, Jenkins y Device Farm.

Fecha de publicación: 2 de febrero de 2016

- [Uso de AWS CodeCommit con repositorios de Git en varias cuentas de Amazon Web Services](#)

Aprenda a clonar su repositorio de CodeCommit y a configurar, con un solo comando, la aplicación ayudante de credenciales para utilizar un rol de IAM específico para las conexiones a ese repositorio.

Fecha de publicación: noviembre de 2015

- [Integración de OpsWorks y AWS CodeCommit](#)

Descubra cómo OpsWorks puede buscar automáticamente aplicaciones y libros de recetas de Chef desde CodeCommit.

Fecha de publicación: 25 de agosto de 2015

- [Uso de AWS CodeCommit y la aplicación ayudante de credenciales de GitHub](#)

Aprenda a configurar el archivo gitconfig para trabajar con los ayudantes de credenciales de CodeCommit y GitHub.

Fecha de publicación: septiembre de 2015

- [Uso de AWS CodeCommit desde Eclipse](#)

Aprenda a utilizar herramientas EGit en Eclipse para trabajar con CodeCommit.

Fecha de publicación: agosto de 2015

- [AWS CodeCommit con credenciales de rol de Amazon EC](#)

Aprenda a utilizar un perfil de instancia de Amazon EC2 al configurar el acceso automatizado de un agente a un repositorio de CodeCommit.

Fecha de publicación: julio de 2015

- [Integración de AWS CodeCommit con Jenkins](#)

Aprenda a utilizar CodeCommit y Jenkins para dar soporte a dos sencillos escenarios de integración continua (CI).

Fecha de publicación: julio de 2015

- [Integración de AWS CodeCommit con Review Board](#)

Aprenda a integrar CodeCommit en un flujo de trabajo de desarrollo con el sistema de revisión de código [Review Board](#).

Fecha de publicación: julio de 2015

Ejemplos de código

Los siguientes son ejemplos de código que pueden interesar a los usuarios de CodeCommit.

- [Mac OS X Script to Periodically Delete Cached Credentials in the OS X Certificate Store](#)

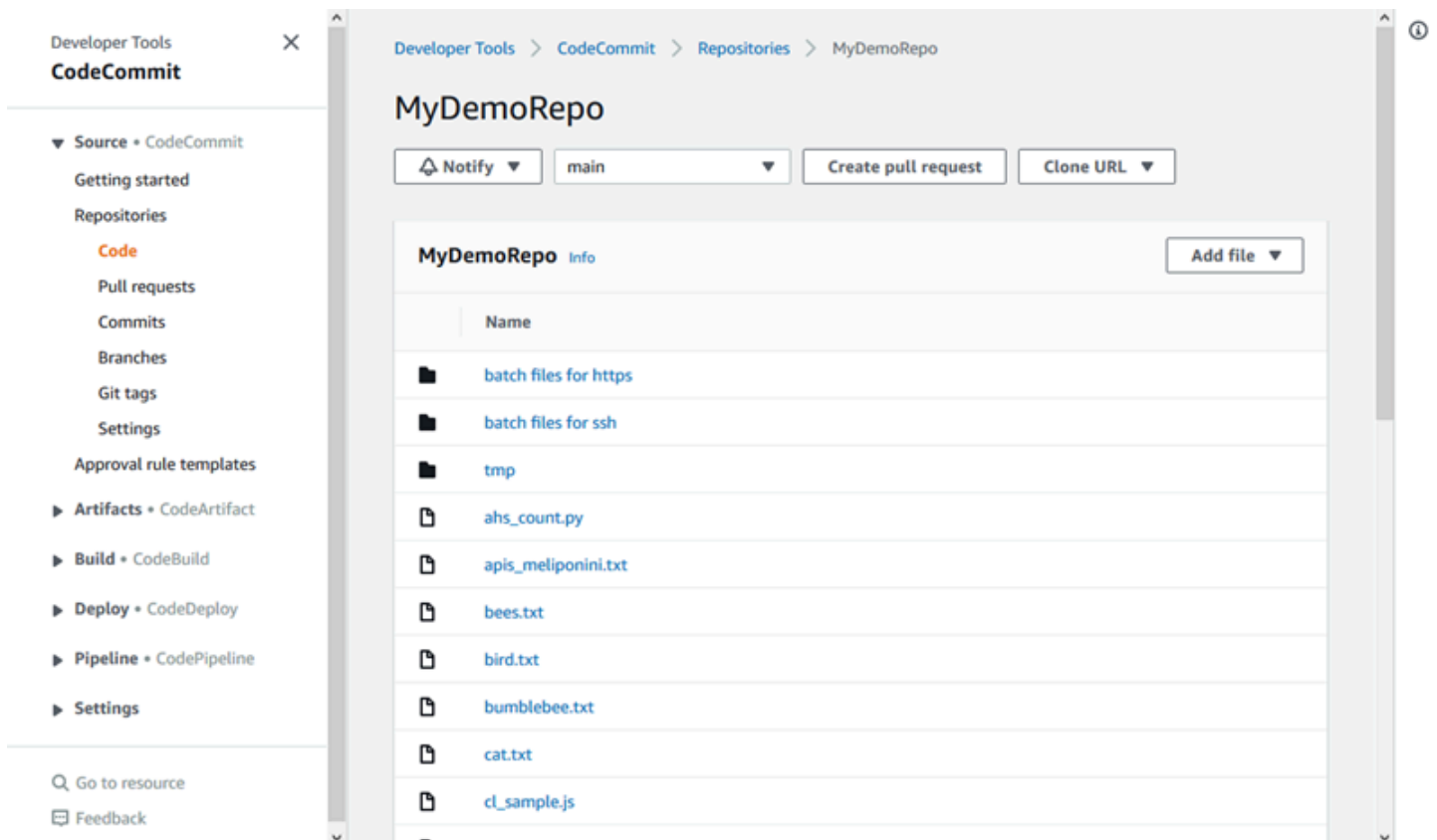
Si utiliza el ayudante de credenciales para CodeCommit en Mac OS X, es posible que esté familiarizado con el problema de las credenciales almacenadas en caché. Este script muestra una solución.

Autor: Nico Coetzee

Fecha de publicación: febrero de 2016

Trabajar con repositorios en AWS CodeCommit

Un repositorio es el objeto de control de la versión fundamental en CodeCommit. En él puede almacenar de manera segura el código y los archivos de su proyecto. También almacena el historial de su proyecto, desde la primera confirmación hasta los últimos cambios. Puede compartir su repositorio con otros usuarios para trabajar juntos en un proyecto. Si agregas AWS etiquetas a los repositorios, puedes configurar las notificaciones para que los usuarios del repositorio reciban correos electrónicos sobre eventos (por ejemplo, otro usuario comentando un código). También puede cambiar la configuración predeterminada para el repositorio, examinar su contenido y mucho más. Puede crear disparadores para su repositorio, de modo que las inserciones de código u otros eventos desencadenen acciones como, por ejemplo, correos electrónicos o funciones del código. Puede incluso configurar un repositorio en el equipo local (un repositorio local) para insertar los cambios en más de un repositorio.



Antes de poder enviar cambios a un CodeCommit repositorio, debe configurar un usuario de IAM en su cuenta de Amazon Web Services o configurar el acceso para el acceso federado o las credenciales temporales. Para obtener más información, consulte [Paso 1: Configuración inicial para](#)

[CodeCommit](#) y [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Para obtener información sobre cómo trabajar con otros aspectos del repositorio en CodeCommit, consulte [Trabajar con archivos](#), [Trabajo con solicitudes de extracción](#), [Uso de confirmaciones](#), [Uso de las ramificaciones](#), y [Trabajo con las preferencias del usuario](#). Para obtener información sobre la migración a CodeCommit, consulte [Migrar a CodeCommit](#).

Temas

- [Cree un AWS CodeCommit repositorio](#)
- [Conéctese a un AWS CodeCommit repositorio](#)
- [Comparte un AWS CodeCommit repositorio](#)
- [Configurar las notificaciones para los eventos en una AWS CodeCommit repositorio](#)
- [Etiquetar repositorios en AWS CodeCommit](#)
- [Gestiona los desencadenantes de un AWS CodeCommit repositorio](#)
- [Amazon CodeGuru Reviewer y AWS CodeCommit](#)
- [Vea los detalles CodeCommit del repositorio](#)
- [Cambio AWS CodeCommit configuración del repositorio](#)
- [Sincronice los cambios entre un repositorio local y un AWS CodeCommit repositorio](#)
- [Envío de confirmaciones a un repositorio de Git adicional](#)
- [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#)
- [Eliminar un AWS CodeCommit repositorio](#)

Cree un AWS CodeCommit repositorio

Utilice la AWS CodeCommit consola o el AWS Command Line Interface (AWS CLI) para crear un CodeCommit repositorio vacío. Para añadir etiquetas a un repositorio después de crearlo, consulte [Añadir una etiqueta a un repositorio](#).

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración](#).

 Note

Según el uso, es posible que se le cobre por crear un repositorio o por acceder a él. Para obtener más información, consulta [los precios](#) en la página de información CodeCommit del producto.

Temas

- [Crear un repositorio \(Consola\)](#)
- [Crea un repositorio \(AWS CLI\)](#)

Crear un repositorio (Consola)

Para crear un CodeCommit repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija la Región de AWS ubicación en la que desea crear el repositorio. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositorios, seleccione Crear repositorio.
4. En la página Create repository (Crear repositorio), en el campo Repository name (Nombre de repositorio), escriba un nombre para el repositorio.

 Note

Los nombres de repositorio distinguen entre mayúsculas y minúsculas. El nombre debe ser único en la Región de AWS y de su cuenta de Amazon Web Services.

5. (Opcional) En Descripción, introduzca una descripción para el repositorio. Esto puede ayudarle a usted y a otros usuarios a conocer el propósito del repositorio.

 Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si es un desarrollador de aplicaciones que utiliza BatchGetRepositories las API GetRepository o las API y planea mostrar el

campo de descripción del repositorio en un navegador web, consulte la [Referencia de CodeCommit API](#).

6. (Opcional) Selecciona Añadir etiqueta para añadir una o más etiquetas de repositorio (una etiqueta de atributo personalizada que te ayuda a organizar y gestionar tus AWS recursos) a tu repositorio. Para obtener más información, consulte [Etiquetar repositorios en AWS CodeCommit](#).
7. (Opcional) Amplíe la configuración adicional para especificar si desea utilizar la clave predeterminada Clave administrada de AWS o la propia clave gestionada por el cliente para cifrar y descifrar los datos de este repositorio. Si decide utilizar su propia clave gestionada por el cliente, debe asegurarse de que esté disponible en el Región de AWS lugar donde vaya a crear el repositorio y de que la clave esté activa. Para obtener más información, consulte [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#).
8. (Opcional) Seleccione Habilitar Amazon CodeGuru Reviewer para Java y Python si este repositorio contiene código Java o Python y desea que CodeGuru Reviewer lo analice. CodeGuru Reviewer utiliza varios modelos de aprendizaje automático para detectar defectos en el código y sugerir mejoras y correcciones en las solicitudes de incorporación de cambios. Para obtener más información, consulte la [Guía del usuario de Amazon CodeGuru Reviewer](#).
9. Seleccione Crear.

Después de crear un repositorio, puedes conectarte a él y empezar a añadir código a través de la CodeCommit consola o de un cliente Git local, o bien integrando tu CodeCommit repositorio con tu IDE favorito. Para obtener más información, consulte [Configuración de AWS CodeCommit](#). También puede añadir su repositorio a una canalización de entrega continua. Para obtener más información, consulte [Guía sobre la canalización simple](#).

Para obtener información sobre el nuevo CodeCommit repositorio, como las URL que se usarán al clonar el repositorio, elige el nombre del repositorio de la lista o simplemente elige el protocolo de conexión que quieres usar junto al nombre del repositorio.

Para compartir este repositorio con otras personas, debe enviarles el enlace HTTPS o SSH a utilizar para clonar el repositorio. Asegúrese de que dispone de los permisos necesarios para acceder al repositorio. Para obtener más información, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Crea un repositorio (AWS CLI)

Puede utilizar el AWS CLI para crear un CodeCommit repositorio. A diferencia de la consola, puede agregar etiquetas a un repositorio si lo crea mediante la AWS CLI.

1. Asegúrese de haber configurado el AWS CLI Región de AWS lugar donde se encuentra el repositorio. Para verificar la región, ejecute el siguiente comando en la línea de comandos o terminal y revise la información del nombre de la región predeterminado.

```
aws configure
```

El nombre de la región Región de AWS por defecto debe coincidir con el del repositorio CodeCommit. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).

2. Ejecute el comando `create-repository`, especificando:

- Un nombre que identifique de forma exclusiva el CodeCommit repositorio (con la `--repository-name` opción).

Note

Este nombre debe ser exclusivo de una cuenta de Amazon Web Services.

- Un comentario opcional sobre el CodeCommit repositorio (con la `--repository-description` opción).
- Un par o varios pares clave-valor opcionales para usarlos como etiquetas en el CodeCommit repositorio (con la `--tags` opción).
- Una clave opcional gestionada por el cliente para usarla al cifrar y descifrar este repositorio. Todos los repositorios se cifran en tránsito y en reposo mediante una clave de en AWS KMS. Si no se especifica ninguna clave, se utiliza la clave AWS `aws/codecommit` gestionada por defecto.

Por ejemplo, para crear un CodeCommit repositorio denominado `MyDemoRepo` con la descripción "My demonstration repository" y una etiqueta con una clave denominada *Team* con el valor de *Saanvi*, utilice este comando.

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-  
description "My demonstration repository" --tags Team=Saanvi
```

 Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si es un desarrollador de aplicaciones que utiliza las BatchGetRepositories API GetRepository o y planea mostrar el campo de descripción del repositorio en un navegador web, consulte la [referencia de la CodeCommit API](#).

3. Si la operación se completó correctamente, el comando devuelve un objeto repositoryMetadata con la siguiente información:
 - La descripción (repositoryDescription)
 - El ID exclusivo generado por el sistema (repositoryId)
 - El nombre (repositoryName)
 - El ID de la cuenta de Amazon Web Services asociada al CodeCommit repositorio (accountId).

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior.

```
{
  "repositoryMetadata": {
    "repositoryName": "MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "lastModifiedDate": 1446071622.494,
    "repositoryDescription": "My demonstration repository",
    "cloneUrlHttp": "https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "defaultBranch": main,
    "kmsKeyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "creationDate": 1446071622.494,
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
    "accountId": "111111111111"
  }
}
```

 Note

Las etiquetas que se agregaron cuando se creó el repositorio no se devuelven en la salida. Para ver una lista de etiquetas asociadas a un repositorio, ejecute el comando [list-tags-for-resource](#).

4. Anote el nombre y el ID del CodeCommit repositorio. Los necesita para monitorear y cambiar la información sobre el CodeCommit repositorio, especialmente si lo usa AWS CLI.

Si olvida el nombre o ID, siga las instrucciones de [Ver los detalles del CodeCommit repositorio \(AWS CLI\)](#).

Después de crear un repositorio, puede conectarlo y empezar a añadir código. Para obtener más información, consulte [Conectarse a un repositorio](#). También puede añadir su repositorio a una canalización de entrega continua. Para obtener más información, consulte [Guía sobre la canalización simple](#).

Conéctese a un AWS CodeCommit repositorio

Cuando te conectas a un CodeCommit repositorio por primera vez, normalmente clonas su contenido en tu máquina local. También puede [añadir y editar archivos](#) en un repositorio directamente desde la CodeCommit consola. De forma alternativa, si ya dispone de un repositorio local, puede agregar un repositorio de CodeCommit remoto. En este tema se proporcionan instrucciones para conectarse a un CodeCommit repositorio. Si desea migrar un repositorio existente a CodeCommit, consulte [Migrar a CodeCommit](#).

 Note

Según el uso, es posible que se le cobre por crear un repositorio o por acceder a él. Para obtener más información, consulte [los precios](#) en la página de información CodeCommit del producto.

Temas

- [Requisitos previos para conectarse a un repositorio CodeCommit](#)
- [Conéctese al CodeCommit repositorio clonando el repositorio](#)

- [Conectar un repositorio local al CodeCommit repositorio](#)

Requisitos previos para conectarse a un repositorio CodeCommit

Antes de poder clonar un CodeCommit repositorio o conectar un repositorio local a un CodeCommit repositorio:

- Debe haber configurado el equipo local con el software y los ajustes necesarios para conectarse a CodeCommit. Esto incluye la instalación y configuración de Git. Para obtener más información, consulte [Configuración](#) y [Primeros pasos con Git y AWS CodeCommit](#).
- Debes tener la URL de clonación del CodeCommit repositorio al que quieres conectarte. Para obtener más información, consulte [Visualización de la información de los repositorios](#).

Si aún no ha creado un CodeCommit repositorio, siga las instrucciones que se indican [Creación de un repositorio](#), copie la URL de clonación del CodeCommit repositorio y vuelva a esta página.

Si tiene un CodeCommit repositorio pero no conoce su nombre, siga las instrucciones que se indican en [Visualización de la información de los repositorios](#).

- Debe tener una ubicación en su máquina local para almacenar una copia local del CodeCommit repositorio al que se conecta. (Esta copia local del CodeCommit repositorio se conoce como repositorio local). A continuación, deberá cambiar a Git y ejecutar los comandos desde dicha ubicación. Por ejemplo, puede utilizar /tmp (para Linux, macOS o Unix) o c:\temp (para Windows) si está creando un clon temporal con fines de prueba. Esta es la ruta de directorio que se usa en estos ejemplos.

Note

Puede usar cualquier directorio que desee. Si va a clonar un repositorio para usarlo durante bastante tiempo, considere la posibilidad de crear el clon desde un directorio de trabajo y no desde uno que se utilice con archivos temporales. Si usa un directorio distinto de /tmp o c:\temp, asegúrese de reemplazarlo por el nuestro cuando siga estas instrucciones.

Conéctese al CodeCommit repositorio clonando el repositorio

Si aún no tiene un repositorio local, siga los pasos de este procedimiento para clonar el CodeCommit repositorio en su máquina local.

1. Cumpla los requisitos previos, incluida la [Configuración](#)

Important

Si no ha completado la instalación, no puede conectarse al repositorio ni clonarlo.

2. Desde el directorio /tmp o el directorio c:\temp, use Git para ejecutar el comando clone. Los siguientes ejemplos muestran cómo clonar un repositorio cuyo nombre pertenece *MyDemoRepo* a la región EE.UU. Este (Ohio).

Para HTTPS usando [credenciales de Git](#) o la aplicación auxiliar de credenciales incluida con la AWS CLI:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Para HTTPS que utilizan [git-remote-codecommit](#), presuponiendo que el perfil es el predeterminado y que la Región de AWS es la que se ha configurado en la AWS CLI:

```
git clone codecommit://MyDemoRepo my-demo-repo
```

Para SSH:

```
git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

En este ejemplo, `git-codecommit.us-east-2.amazonaws.com` es el punto de conexión de Git para la región EE.UU. Este (Ohio) donde se encuentra el repositorio, `MyDemoRepo` representa el nombre de tu CodeCommit repositorio y `my-demo-repo` representa el nombre del directorio que Git crea en el /tmp directorio o el c:\temp directorio. Para obtener más información sobre Regiones de AWS esa compatibilidad CodeCommit y las conexiones de Git para ellas Regiones de AWS, consulte [Regiones y puntos de conexión de Git](#).

 Note

Al usar SSH en los sistemas operativos Windows para clonar un repositorio, es posible que necesite agregar el ID de la clave SSH a la cadena de conexión de la siguiente manera:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Para obtener más información, consulte [Para conexiones SSH en Windows](#) y [Solución de problemas](#).

Una vez que Git crea el directorio, extrae una copia de tu CodeCommit repositorio en el directorio recién creado.

Si el CodeCommit repositorio es nuevo o está vacío por algún motivo, verás un mensaje en el que se indica que estás clonando un repositorio vacío. Esto es normal.

 Note

Si recibes un error que indica que Git no encuentra el CodeCommit repositorio o que no tienes permiso para conectarte al CodeCommit repositorio, asegúrate de cumplir los [requisitos previos](#), incluida la asignación de permisos al usuario de IAM y la configuración de tus credenciales de usuario de IAM para Git y CodeCommit en la máquina local. Además, asegúrese de haber especificado el nombre del repositorio correcto.

Una vez que hayas conectado correctamente tu repositorio local a tu CodeCommit repositorio, ya estás listo para empezar a ejecutar comandos de Git desde el repositorio local para crear confirmaciones, ramas y etiquetas y enviar y extraer del CodeCommit repositorio.

Conectar un repositorio local al CodeCommit repositorio

Complete los pasos siguientes si ya dispone de un repositorio local y desea agregar un repositorio de CodeCommit como repositorio remoto. Si ya tienes un repositorio remoto y quieres enviar tus

confirmaciones a CodeCommit ese otro repositorio remoto, sigue los pasos que se indican. [Envío de confirmaciones a dos repositorios](#)

1. Complete los [requisitos previos](#).
2. Desde la línea de comandos o desde la terminal, cambia al directorio de repositorios local y ejecuta el `git remote add` comando para añadir el CodeCommit repositorio como repositorio remoto para tu repositorio local.

Por ejemplo, el siguiente comando añade el repositorio denominado **origin** a `https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo`:

Para HTTPS:

```
git remote add origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Para SSH:

```
git remote add origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Este comando no devuelve nada.

3. Para comprobar que has añadido el CodeCommit repositorio como remoto a tu repositorio local, ejecuta el `git remote -v` comando, que debería generar un resultado similar al siguiente:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para SSH:

```
origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Una vez que hayas conectado correctamente tu repositorio local a tu CodeCommit repositorio, estarás listo para empezar a ejecutar comandos de Git desde el repositorio local para crear confirmaciones, ramas y etiquetas, y para insertar y extraer del CodeCommit repositorio.

Comparte un AWS CodeCommit repositorio

Una vez creado un CodeCommit repositorio, puede compartirlo con otros usuarios. En primer lugar, determine si va a utilizar un acceso federado, credenciales temporales o un proveedor de identidad web, como el Centro de identidades de IAM, al acceder CodeCommit, o si quiere utilizar credenciales de Git o pares de claves SSH con los usuarios de IAM. Si utiliza el primero, tendrá que configurar los usuarios, el acceso y los permisos para su proveedor de identidad y, a continuación, proporcionar instrucciones a los usuarios para que usen `git-remote-codecommit`. Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con `git-remote-codecommit`](#) y [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#).

No puede usar credenciales de Git ni pares de claves SSH con proveedores de identidad o acceso federado, pero muchos IDE funcionan mejor con estas credenciales. En este caso, decida qué protocolo (HTTPS o SSH) recomendará a los usuarios en la clonación y el uso de un cliente de Git o un IDE para conectarse a su repositorio. A continuación, envíe la dirección URL y la información de conexión a los usuarios con los que desee compartir el repositorio. Dependiendo de sus requisitos de seguridad, compartir un repositorio también podría requerir la creación de un grupo de IAM, la aplicación de políticas administradas a ese grupo y la edición de las políticas de IAM para refinar el acceso o crear y usar roles de IAM.

Note

Una vez que haya concedido a los usuarios acceso de consola al repositorio, estos pueden añadir o editar archivos directamente en la consola sin tener que configurar un cliente de Git u otro tipo de conexión. Para obtener más información, consulte [Crear o añadir un archivo a un repositorio de AWS CodeCommit](#) y [Editar el contenido de un archivo en un AWS CodeCommit repositorio](#).

Estas instrucciones se han redactado presuponiendo que ya ha completado los pasos indicados en [Configuración](#) y [Creación de un repositorio](#).

 Note

Según el uso, es posible que se le cobre por crear un repositorio o por acceder a él. Para obtener más información, consulta los [precios](#) en la página de información del CodeCommit producto.

Temas

- [Seleccione el protocolo de conexión para compartir con sus usuarios](#)
- [Creación de políticas de IAM para su repositorio](#)
- [Creación de un grupo IAM para usuarios de repositorios](#)
- [Compartir la información de conexión con los usuarios](#)

Seleccione el protocolo de conexión para compartir con sus usuarios

Al crear un repositorio en CodeCommit, se generan dos puntos de conexión: uno para las conexiones HTTPS y otro para las conexiones SSH. Ambas proporcionan conexiones seguras a través de una red. Los usuarios pueden utilizar cualquiera de los dos protocolos. Ambos extremos permanecen activos, independientemente del protocolo que recomiende a los usuarios.

Las conexiones HTTPS requieren:

- Credenciales de Git, que los usuarios de IAM pueden generar por sí mismos en IAM. Las credenciales de Git son el método más sencillo de configurar y usar para los usuarios.
- Una clave de AWS acceso o un rol que deben asumir, que los usuarios del repositorio deben configurar en su perfil de credenciales. Puede configurar git-remote-codecommit (recomendado) o la aplicación ayudante de credenciales incluida en la AWS CLI. Estos son los únicos métodos disponibles para la cuenta raíz o los usuarios federados.

Las conexiones SSH requieren que los usuarios:

- Generen un par de clave pública-privada.
- Guarden la clave pública.
- Asocien la clave pública con su usuario de IAM.
- Configuren su archivo de hosts conocidos en su equipo local.

- Creen y mantengan un archivo de configuración en sus equipos locales.

Como se trata de un proceso de configuración más complejo, le recomendamos que elija las credenciales HTTPS y Git para las conexiones CodeCommit.

Para obtener más información sobre HTTPS, SSH, Git, git-remote-codecommit y los repositorios remotos, consulte [Configuración](#), [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#) o la documentación de Git.

Note

Aunque Git es compatible con una gran variedad de protocolos de conexión, CodeCommit no admite conexiones con protocolos no seguros, como el protocolo local o HTTP genérico.

Creación de políticas de IAM para su repositorio

AWS proporciona tres políticas gestionadas en IAM para CodeCommit. Estas políticas no se pueden editar y afectan a todos los repositorios asociados con su cuenta de Amazon Web Services. Sin embargo, puede utilizar estas políticas como plantillas para crear sus propias políticas administradas personalizadas, que se aplican únicamente al repositorio que quiera compartir. Su política administrada por el cliente puede afectar específicamente al repositorio que quiera compartir. Para obtener más información, consulte [Políticas administradas](#) y [Usuarios y grupos de IAM](#).

Tip

Si desea un control más específico sobre el acceso a su repositorio, puede crear más de una política administrada por el cliente y aplicar las políticas a diferentes usuarios de IAM y grupos.

Para obtener más información acerca de cómo revisar el contenido de las políticas administradas y utilizar políticas para crear y aplicar permisos, consulte [Autenticación y control de acceso para AWS CodeCommit](#).

Creación de una política administrada por el cliente para su repositorio

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>

2. En el área de navegación Dashboard, elija Políticas y, a continuación, seleccione Create Policy.
3. En la página Crear política, seleccione Importar política administrada.
4. En la página Importar políticas administradas, en Filtrar políticas, introduzca **AWSCodeCommitPowerUser**. Seleccione el botón situado junto al nombre de la política y, a continuación, seleccione Importar.
5. En la página Crear política, elija JSON. Sustituya la parte «*» de la Resource línea correspondiente a CodeCommit las acciones por el nombre de recurso de Amazon (ARN) del CodeCommit repositorio, como se muestra a continuación:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

 Tip

Para buscar el ARN del CodeCommit repositorio, vaya a la CodeCommit consola, elija el nombre del repositorio de la lista y, a continuación, elija Configuración. Para obtener más información, consulte [Visualización de la información de los repositorios](#).

Si desea que esta política se aplique a más de un repositorio, añada cada repositorio como recurso y especifique su ARN. Añada una coma entre cada declaración de recursos, tal y como se muestra aquí:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Cuando haya terminado de editar, seleccione Revisar la política.

6. En la página de revisión de la política, en Nombre, introduzca un nombre nuevo para la política (por ejemplo, **AWSCodeCommitPowerUser-MyDemoRepo**). Opcionalmente, puede proporcionar una descripción opcional para esta política.
7. Seleccione Crear política.

Creación de un grupo IAM para usuarios de repositorios

Para administrar el acceso a su repositorio, cree un grupo para los usuarios de IAM, añada usuarios de IAM a este grupo y adjunte la política administrada por el cliente que haya creado en el paso anterior. Como alternativa, puede crear un rol con una política administrada por el cliente adjunta y hacer que los usuarios asuman ese rol.

Si usa SSH, debe adjuntar otra política administrada al IAMUserSSHKeys grupo, la política administrada de IAM que permite a los usuarios cargar su clave pública de SSH y asociarla al usuario de IAM al que utilizan para conectarse. CodeCommit

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
2. En el área de navegación Dashboard, elija Groups y, a continuación, seleccione Create New Group.
3. En la página Set Group Name (Establecer nombre del grupo), en Group Name (Nombre del grupo), introduzca un nombre para el grupo (por ejemplo, *MyDemoRepoGroup*) y seleccione Next Step (Siguiente paso). Tenga en cuenta la posibilidad de incluir el nombre del repositorio como parte del nombre del grupo.

Note

Este nombre debe ser exclusivo de una cuenta de Amazon Web Services.

4. Selecciona la casilla situada junto a la política gestionada por el cliente que creaste en la sección anterior (por ejemplo, AWSCodeCommitPowerUser-MyDemoRepo).
5. En la página Review, elija Create Group. IAM crea este grupo con las políticas especificadas ya adjuntas. El grupo aparece en la lista de grupos asociados con su cuenta de Amazon Web Services.
6. Seleccione su grupo de la lista.
7. En la página de resumen de grupo, elija la pestaña Users y, continuación, seleccione Add Users to Group. En la lista que muestra todos los usuarios asociados a tu cuenta de Amazon Web Services, selecciona las casillas situadas junto a los usuarios a los que quieres permitir el acceso al CodeCommit repositorio y, a continuación, selecciona Añadir usuarios.

 Tip

Puede usar el cuadro de búsqueda para encontrar rápidamente a los usuarios por su nombre.

8. Cuando haya agregado sus usuarios, cierre la consola de IAM.

Compartir la información de conexión con los usuarios

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositories (Repositorios), elija el repositorio que desea compartir.
4. En Clone URL (URL de clonación), elija el protocolo que desee que utilicen los usuarios. Esto copiará la URL de clonación del protocolo de conexión.
5. Envía a tus usuarios la URL del clon junto con cualquier otra instrucción, como instalar AWS CLI, configurar un perfil o instalar Git. Asegúrese de incluir la información de configuración para el protocolo de conexión (por ejemplo, HTTPS).

El siguiente ejemplo de correo electrónico proporciona información para los usuarios que se conectan al MyDemoRepo repositorio con el protocolo de conexión HTTPS y las credenciales de Git en la región EE.UU. Este (Ohio) (us-east-2). Este correo electrónico se ha escrito partiendo del supuesto de que el usuario ya ha instalado Git y sabe cómo utilizarlo.

I've created a CodeCommit repository for us to use while working on our project. The name of the repository is *MyDemoRepo*, and it is in the US East (Ohio) (us-east-2) region. Here's what you need to do in order to get started using it:

1. Make sure that your version of Git on your local computer is 1.7.9 or later.
2. Generate Git credentials for your IAM user by signing into the IAM console here: <https://console.aws.amazon.com/iam/>.

Switch to the **Security credentials** tab for your IAM user and choose the **Generate** button in **HTTPS Git credentials for CodeCommit**.

Make sure to save your credentials in a secure location!

3. Switch to a directory of your choice and clone the CodeCommit repository to your local machine by running the following command:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```
4. When prompted for user name and password, use the Git credentials you just saved.

That's it! If you'd like to learn more about using CodeCommit, you can start with the tutorial [here](#).

Puede encontrar instrucciones de configuración completas en [Configuración](#).

Configurar las notificaciones para los eventos en una AWS CodeCommit repositorio

Puede configurar reglas de notificación en un repositorio para que los usuarios de dicho repositorio reciban mensajes de correo electrónico sobre los tipos de eventos que se especifiquen. Las notificaciones se envían cuando los eventos coinciden con la configuración de la regla de notificación. Puede crear un tema de Amazon SNS que se puede utilizar para las notificaciones o bien utilizar uno existente en su cuenta de Amazon Web Services. Puede utilizar la CodeCommit consola y el AWS CLI para configurar las reglas de notificación.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Settings

Create notification rule

Notification rules set up a subscription to events that happen with your resources. When these events occur, you will receive notifications sent to the targets you designate. You can manage your notification preferences in Settings. [Info](#)

Notification rule settings

Notification name

My notification rule for MyDemoRepo

Detail type

Choose the level of detail you want in notifications. [Learn more about notifications and security](#)

☒ Full

Includes any supplemental information about events provided by the resource or the notifications feature.

☐ Basic

Includes only information provided in resource events.

Events that trigger notifications

Select none

Select all

Comments

☐ On Commits

☒ On Pull requests

Pull request

☒ Source Updated

☒ Created

☒ Status Changed

☒ Merged

Branches and tags

☐ Created

☒ Deleted

☐ Updated

Targets

Choose an SNS topic to use as the target for the notification rule. Users can subscribe to the notification topic to receive emails about events. You can also configure integration between the SNS topic and AWS Chatbot, so that users receive notifications in Slack channels or Amazon Chime chatrooms.

You can also configure integration between the SNS topic and AWS Chatbot, so that users receive notifications in Slack channels or Amazon Chime chatrooms. [Learn more](#)

Amazon SNS topic ARN

arn:aws:sns:us-east-2:111111111111:codestar-notifications-MyTopicForMyDe X

Remove target

Add target

Create SNS topic

Cancel

Submit

Temas

- [Uso de reglas de notificación de repositorios](#)
- [Creación de una regla de notificación](#)

- [Cambiar o deshabilitar notificaciones](#)
- [Eliminación de notificaciones](#)

Uso de reglas de notificación de repositorios

Configurar notificaciones sirve de ayuda a los usuarios del repositorio, ya que pueden recibir correos electrónicos cuando alguien realiza una acción que afecta a otro usuario. Por ejemplo, puede configurar una regla de notificación para que se envíen un mensaje cuando se realicen comentarios sobre una confirmación. En esta configuración, cuando un usuario del repositorio comenta una línea de código en una confirmación, los demás usuarios del repositorio reciben un mensaje de correo electrónico. Pueden iniciar sesión y ver el comentario. Las respuestas a los comentarios también generan correos electrónicos, lo que permite a los usuarios del repositorio mantenerse informados.

Las reglas de notificación son diferentes de las de los activadores del repositorio y también son diferentes de las notificaciones que podías configurar en la CodeCommit consola antes del 5 de noviembre de 2019.

- Aunque puede configurar un disparador para que utilice Amazon SNS para enviar mensajes de correo electrónico sobre algunos eventos del repositorio, esos eventos se limitan a eventos operativos como, por ejemplo, la creación de ramificaciones y el envío de código a una ramificación. Los desencadenadores no utilizan las reglas de CloudWatch eventos para evaluar los eventos del repositorio. Su ámbito es más limitado. Para obtener más información acerca del uso de disparadores, consulte [Administración de los disparadores de un repositorio](#).
- Las notificaciones configuradas antes del 5 de noviembre de 2019 tenían menos tipos de eventos disponibles y no podían configurarse para integrarlas con salas de chat de Amazon Chime o canales de Slack. Puede seguir utilizando las notificaciones configuradas antes del 5 de noviembre de 2019, pero no puede crear más notificaciones de este tipo. En su lugar, cree y utilice reglas de notificación. Le recomendamos que utilice reglas de notificación y deshabilite o elimine las notificaciones creadas antes del 5 de noviembre de 2019. Para obtener más información, consulte [Creación de una regla de notificación](#) y [Eliminación de notificaciones](#).

Creación de una regla de notificación

Puede utilizar reglas de notificación para notificar a los usuarios los cambios importantes; por ejemplo, si se crea una solicitud de extracción en un repositorio. Las reglas de notificación

especifican tanto los eventos como el tema de Amazon SNS que se utiliza para enviar notificaciones. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

 Note

Esta característica no está disponible en la región de Europa (Milán). Para obtener información sobre cómo configurar las notificaciones en la experiencia disponible en esa región, consulte [Configurar las notificaciones del repositorio](#).

Puede usar la consola o la AWS CLI para crear reglas de notificación para AWS CodeCommit.

Para crear una regla de notificación (consola)

1. Inicie sesión en Consola de administración de AWS y abra la CodeCommit consola en <https://console.aws.amazon.com/codecommit/>.
2. Elija Repositories (Repositorios) y seleccione el repositorio en el que desea añadir reglas de notificación.
3. En la página del repositorio, seleccione Notify (Notificar) y Create notification rule (Crear regla de notificación). También puede ir a la página Settings (Configuración) del repositorio y seleccionar Create notification rule (Crear regla de notificación).
4. En Nombre de la notificación, introduzca un nombre para la regla.
5. En Tipo de detalle, selecciona Básico si quieres que solo se EventBridge incluya en la notificación la información proporcionada a Amazon. Selecciona Completa si deseas incluir la información proporcionada a Amazon EventBridge y la información que pueda proporcionar el administrador de notificaciones CodeCommit o el administrador de notificaciones.

Para obtener más información, consulte [Descripción del contenido y la seguridad de las notificaciones](#).

6. En Eventos que activan notificaciones, seleccione los eventos para los que desea enviar notificaciones. Para obtener más información, consulte [Eventos para reglas de notificación en repositorios](#).
7. En Destinos, realice una de las siguientes operaciones:
 - Si ya ha configurado un recurso para utilizarlo con notificaciones, en Elegir tipo de destino, elija Amazon Q Developer en aplicaciones de chat (Slack) o Tema de SNS. En Elegir

destino, elija el nombre del cliente (para un cliente de Slack configurado en Amazon Q Developer en aplicaciones de chat) o el nombre de recurso de Amazon (ARN) del tema de Amazon SNS (para los temas de Amazon SNS ya configurados con la política necesaria para las notificaciones).

- Si no ha configurado un recurso para utilizarlo con notificaciones, elija Crear destino y, a continuación, elija Tema de SNS. Indique el nombre del tema después de codestar-notifications- y, a continuación, elija Crear.

Note

- Si crea el tema de Amazon SNS durante la creación de la regla de notificación, se aplica la política que permite a la característica de notificaciones publicar eventos en el tema. El uso de un tema creado para las reglas de notificación lo ayuda a garantizar que solo suscriba a los usuarios que desea recibir notificaciones sobre este recurso.
- No se puede crear un cliente de Amazon Q Developer en aplicaciones de chat durante la creación de una regla de notificación. Si elige Amazon Q Developer en aplicaciones de chat (Slack), aparecerá un botón que lo llevará a configurar un cliente en Amazon Q Developer en aplicaciones de chat. Elija esa opción para abrir la consola de Amazon Q Developer en aplicaciones de chat. Para obtener más información, consulte [Configuración de la integración entre las notificaciones y Amazon Q Developer en aplicaciones de chat](#).
- Si desea utilizar un tema de Amazon SNS ya existente como destino, debe agregar la política necesaria para las notificaciones de AWS CodeStar además de otras políticas que puedan existir para ese tema. Para obtener más información, consulte [Configuración de temas de Amazon SNS para notificaciones](#) y [Descripción del contenido y la seguridad de las notificaciones](#).

8. Para terminar de crear la regla, elija Enviar.
9. Debe suscribir a los usuarios al tema de Amazon SNS de la regla antes de que puedan recibir notificaciones. Para obtener más información, vea [Suscribir usuarios a temas de Amazon SNS que son destinos](#). También puede configurar la integración entre notificaciones y Amazon Q Developer en aplicaciones de chat para enviar notificaciones a salas de chat de Amazon Chime. Para obtener más información, consulte [Configuración de la integración entre notificaciones y Amazon Q Developer en aplicaciones de chat](#).

Para crear una regla de notificación (AWS CLI)

1. En un terminal o símbolo del sistema, ejecute el comando `create-notification rule` para generar el esqueleto JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

Puede asignar al archivo el nombre que desee. En este ejemplo, el archivo se denomina *rule.json*.

2. Abra el archivo JSON en un editor de texto sin formato y edítelo para incluir el recurso, los tipos de eventos y el destino que desea para la regla. El siguiente ejemplo muestra una regla de notificación con **MyNotificationRule** el nombre de un repositorio nombrado *MyDemoRepo* en AWS una cuenta con el ID *123456789012*. Las notificaciones con todos los detalles se envían a un tema de Amazon SNS denominado *MyNotificationTopic* cuando se crean las sucursales y las etiquetas:

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeIds": [  
    "codecommit-repository-branches-and-tags-created"  
  ],  
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",  
  "Targets": [  
    {  
      "TargetType": "SNS",  
      "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL"  
}
```

Guarde el archivo.

3. Mediante el archivo que acaba de modificar, en el terminal o línea de comandos, vuelva a ejecutar el comando `create-notification-rule` para crear la regla de notificación:

```
aws codestar-notifications create-notification-rule --cli-input-json
file://rule.json
```

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

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

Cambiar o deshabilitar notificaciones

Puedes usar la AWS CodeCommit consola para cambiar la configuración de las notificaciones creadas antes del 5 de noviembre de 2019, incluidos los tipos de eventos que envían correos electrónicos a los usuarios y el tema Amazon SNS que se usa para enviar correos electrónicos sobre el repositorio. También puede utilizar la consola de CodeCommit para administrar la lista de direcciones de correo electrónico y los puntos de enlace suscritos al tema o para deshabilitar las notificaciones.

Cómo cambiar la configuración de las notificaciones

1. Abra la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositories (Repositorios), elija el nombre del repositorio donde desea configurar las notificaciones creadas antes del 5 de noviembre de 2019.
3. En el panel de navegación, elija Settings y luego elija Notifications. Si aparece un banner que le informa de que tiene notificaciones en lugar de reglas de notificación, elija Manage existing notifications (Administrar notificaciones existentes).
4. Elija Edit (Edición de).
5. Realice sus cambios y, a continuación, elija Save.

Deshabilitar las notificaciones es una forma sencilla de impedir temporalmente que los usuarios reciban mensajes de correo electrónico sobre eventos del repositorio.

Para eliminar permanentemente una notificación creada antes del 5 de noviembre de 2019, siga los pasos que se indican en [Eliminación de notificaciones](#).

Para deshabilitar las notificaciones

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio donde desea deshabilitar las notificaciones.
3. En el panel de navegación, elija Settings y luego elija Notifications. Seleccione Manage existing notifications (Administrar notificaciones existentes).
4. Elija Edit (Editar) y, en Event status (Estado del evento), utilice el control deslizante para desactivar Enable notifications (Habilitar notificaciones). Seleccione Save.
5. El estado del evento cambia a Disabled (Deshabilitado). No se envían mensajes de correo electrónico sobre los eventos. Al deshabilitar las notificaciones, la regla de CloudWatch eventos del repositorio se deshabilita automáticamente. No cambie su estado manualmente en la consola de CloudWatch eventos.

Eliminación de notificaciones

Si ya no quieres usar las notificaciones creadas para un repositorio antes del 5 de noviembre de 2019, puedes eliminar la regla de Amazon CloudWatch Events asociada a la notificación. De este modo, la notificación se eliminará automáticamente. No se eliminan las suscripciones ni el tema de Amazon SNS que se utiliza para las notificaciones.

Note

Si cambia el nombre de un repositorio desde la consola, las notificaciones creadas antes el 5 de noviembre de 2019 seguirán funcionando sin modificaciones. Sin embargo, si cambia el nombre del repositorio desde la línea de comandos o mediante la API, las notificaciones dejan de funcionar. La forma más sencilla de restablecer las notificaciones consiste en eliminar la configuración de las notificaciones y, a continuación, volver a configurarla.

Para eliminar la configuración de las notificaciones

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio del que desea eliminar las notificaciones creadas antes del 5 de noviembre de 2019.

3. En el panel de navegación, elija Settings y luego elija Notifications. Si aparece un banner que le informa de que tiene notificaciones en lugar de reglas de notificación, elija Manage existing notifications (Administrar notificaciones existentes).
4. En la regla de CloudWatch eventos, copia el nombre de la regla que se creó para la notificación.
5. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
6. En Events (Eventos), elija Rules (Reglas). En Name (Nombre), pegue el nombre de la regla creada para la notificación. Elija la regla y, en Actions (Acciones) seleccione Delete (Eliminar).
7. (Opcional) Para cambiar o eliminar el tema de Amazon SNS utilizado para las notificaciones después de eliminar la configuración de notificaciones, vaya a la consola de Amazon SNS en <https://console.aws.amazon.com/sns/v3/home> Para obtener más información, consulte [Limpiar](#) en la [Guía para desarrolladores de Amazon Simple Notification Service](#).

Etiquetar repositorios en AWS CodeCommit

Una etiqueta es una etiqueta de atributo personalizada que usted o AWS asigna a un AWS recurso. AWS las etiquetas son diferentes de las etiquetas de Git, que se pueden aplicar a las confirmaciones. Cada AWS etiqueta consta de dos partes:

- Una clave de etiqueta (por ejemplo, CostCenter, Environment, Project o Secret). Las claves de etiquetas distinguen entre mayúsculas y minúsculas.
- Un campo opcional que se denomina valor de etiqueta (por ejemplo, 111122223333, Production o el nombre de un equipo). Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía. Al igual que las claves de etiquetas, los valores de las etiquetas distinguen mayúsculas de minúsculas.

En conjunto, se conocen como pares clave-valor. Para conocer los límites del número de etiquetas que puede tener en un repositorio y las restricciones sobre las claves y valores de las etiquetas, consulte [Límites](#).

Las etiquetas ayudan a identificar y organizar AWS los recursos. Muchos AWS servicios admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a un CodeCommit repositorio que a un bucket de Amazon S3. Para obtener más información sobre las estrategias de etiquetado, consulte [Tagging AWS](#) Resources.

En CodeCommit, el recurso principal es un repositorio. Puede usar la CodeCommit consola AWS CLI, las CodeCommit API o AWS los SDK para agregar, administrar y eliminar etiquetas de un repositorio. Además de utilizar etiquetas para identificar, organizar y realizar el seguimiento de del repositorio, puede utilizarlas en las políticas de IAM para ayudar a controlar quién puede ver e interactuar con el repositorio. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Temas

- [Añadir una etiqueta a un repositorio](#)
- [Visualización de etiquetas para un repositorio](#)
- [Editar etiquetas de un repositorio](#)
- [Eliminar una etiqueta de un repositorio](#)

Añadir una etiqueta a un repositorio

Añadir etiquetas a un repositorio puede ayudarte a identificar y organizar tus AWS recursos y a gestionar el acceso a ellos. En primer lugar, añada una o más etiquetas (pares clave-valor) a un repositorio. Tenga en cuenta que hay límites en el número de etiquetas que puede tener en un repositorio. Existen restricciones sobre los caracteres que se pueden utilizar en los campos de clave y valor. Para obtener más información, consulte [Límites](#). Después de tener etiquetas, puede crear políticas de IAM para administrar el acceso al repositorio basándose en estas etiquetas. Puedes usar la CodeCommit consola o la AWS CLI para añadir etiquetas a un repositorio.

Important

Añadir etiquetas a un repositorio puede afectar al acceso a dicho repositorio. Antes de añadir una etiqueta a un repositorio, asegúrese de revisar las políticas de IAM que puedan utilizar etiquetas para controlar el acceso a recursos como los repositorios. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Para obtener más información sobre cómo añadir etiquetas a un repositorio al crearlo, consulte [Crear un repositorio \(Consola\)](#).

Temas

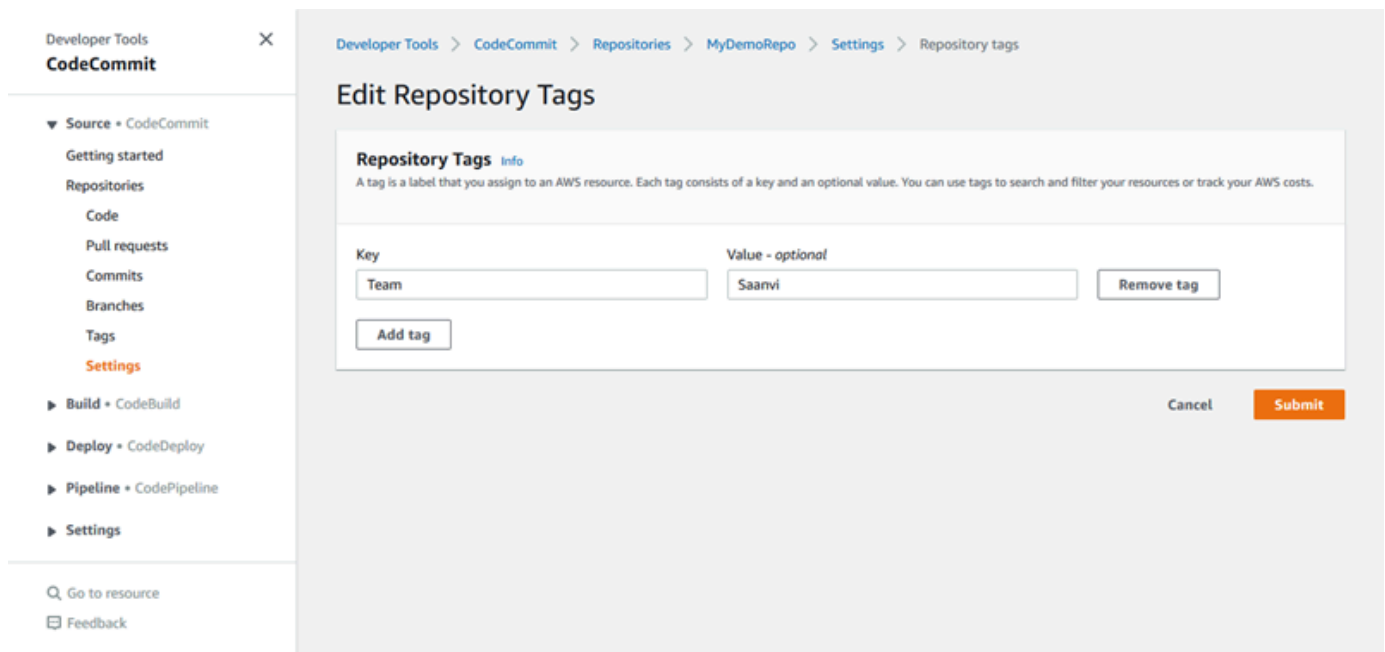
- [Añadir una etiqueta a un repositorio \(consola\)](#)

- [Añadir una etiqueta a un repositorio \(AWS CLI\)](#)

Añadir una etiqueta a un repositorio (consola)

Puede utilizar la CodeCommit consola para añadir una o más etiquetas a un CodeCommit repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio al que desea añadir etiquetas.
3. En el panel de navegación, seleccione Configuración. Elija Repository tags (Etiquetas de repositorio).
4. Si no se han añadido etiquetas al repositorio, seleccione Add tag (Añadir etiqueta). De lo contrario, elija Edit (Editar) y, a continuación, seleccione Add tag (Añadir etiqueta).
5. En Clave, escriba un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.



6. (Opcional) Para añadir otra etiqueta, vuelva a elegir Add tag (Añadir etiqueta).
7. Cuando haya terminado de añadir etiquetas, seleccione Submit (Enviar).

Añadir una etiqueta a un repositorio (AWS CLI)

Siga estos pasos para utilizar el AWS CLI para añadir una etiqueta a un CodeCommit repositorio. Para agregar una etiqueta a un repositorio al crearlo, consulte [Crea un repositorio \(AWS CLI\)](#).

En estos pasos, se presupone que ya ha instalado una versión reciente de la AWS CLI o que la ha actualizado a la versión actual. Para obtener más información, consulte [Instalación de la AWS Command Line Interface](#).

En el terminal o en la línea de comandos, ejecute el comando `tag-resource`, especificando el nombre de recurso de Amazon (ARN) del repositorio al que desea agregar etiquetas y la clave y el valor de la etiqueta que desea agregar. Puede agregar más de una etiqueta a un repositorio. Por ejemplo, para etiquetar un repositorio *MyDemoRepo* con dos etiquetas, una clave de etiqueta *Status* con el valor de etiqueta de *Secret* y una clave de etiqueta *Team* con el valor de etiqueta de *Saanvi*:

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Status=Secret,Team=Saanvi
```

Si se ejecuta correctamente, este comando no devuelve nada.

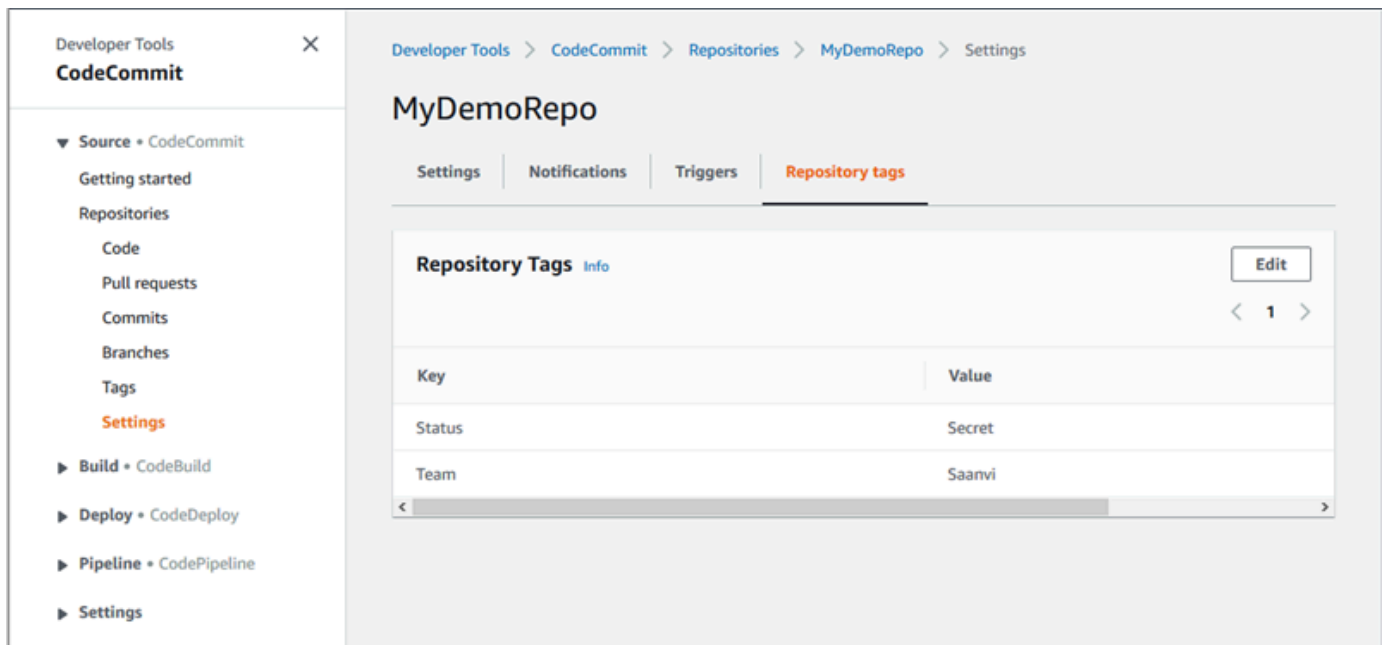
Visualización de etiquetas para un repositorio

Las etiquetas pueden ayudarte a identificar y organizar tus AWS recursos y a administrar el acceso a ellos. Para obtener más información sobre las estrategias de etiquetado, consulte [Etiquetado AWS de recursos](#). Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Ver etiquetas de un repositorio (consola)

Puedes usar la CodeCommit consola para ver las etiquetas asociadas a un CodeCommit repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas etiquetas desea ver.
3. En el panel de navegación, seleccione Configuración. Elija Repository tags (Etiquetas de repositorio).



Ver las etiquetas de un repositorio (AWS CLI)

Siga estos pasos para usar el AWS CLI para ver las AWS etiquetas de un CodeCommit repositorio. Si no se han añadido etiquetas, la lista obtenida está vacía.

En el terminal o la línea de comandos, ejecute el comando `list-tags-for-resource`. Por ejemplo, para ver una lista de las claves de etiqueta y valores de etiquetas para un repositorio denominado *MyDemoRepo* con el ARN *arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo*:

```
aws codecommit list-tags-for-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "Saanvi"
  }
}
```

Editar etiquetas de un repositorio

Puede cambiar el valor de una etiqueta asociada a un repositorio. También puede cambiar el nombre de la clave, lo que equivale a eliminar la etiqueta actual y añadir otra distinta con el nuevo nombre y el mismo valor que la otra clave. Tenga en cuenta que hay límites en los caracteres que puede utilizar en los campos de clave y valor. Para obtener más información, consulte [Límites](#).

Important

Editar las etiquetas de un repositorio puede afectar al acceso a dicho repositorio. Antes de editar el nombre (clave) o valor de una etiqueta de un repositorio, asegúrese de revisar cualquier política de IAM que pueda usar la clave o el valor de una etiqueta para controlar el acceso a recursos como los repositorios. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Editar una etiqueta de un repositorio (consola)

Puede utilizar la CodeCommit consola para editar las etiquetas asociadas a un CodeCommit repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas etiquetas desea editar.
3. En el panel de navegación, seleccione Configuración. Elija Repository tags (Etiquetas de repositorio).
4. Elija Edit (Edición de).

5.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Settings > Repository tags

Edit Repository Tags

Repository Tags Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Each resource can have up to 50 tags. Keys cannot begin with "AWS:".

Key	Value - optional	
Status	Secret	Remove tag
Team	Saanvi	Remove tag

Add tag

Cancel Submit

Realice una de las siguientes acciones:

- Para cambiar la etiqueta, escriba un nuevo nombre en Key (Clave). Cambiar el nombre de la etiqueta equivale a quitar una etiqueta y añadir otra nueva con el nuevo nombre de clave.
- Para cambiar el valor de una etiqueta, escriba un nuevo valor. Si desea cambiar el valor por nada, elimine el valor actual y deje el campo en blanco.

6. Cuando haya terminado de editar las etiquetas, seleccione Submit (Enviar).

Editar las etiquetas de un repositorio (AWS CLI)

Siga estos pasos para usar el AWS CLI para actualizar una etiqueta de un CodeCommit repositorio. Puede cambiar el valor de una clave existente o añadir otra clave.

En el terminal o en la línea de comandos, ejecute el comando `tag-resource`, especificando el nombre de recurso de Amazon (ARN) del repositorio donde desea actualizar una etiqueta, y especifique la clave y el valor de la etiqueta:

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Team=Li
```

Eliminar una etiqueta de un repositorio

Puede eliminar una o más etiquetas asociadas a un repositorio. Al eliminar una etiqueta, no se elimina la etiqueta de otros AWS recursos que estén asociados a esa etiqueta.

Important

Eliminar las etiquetas de un repositorio puede afectar al acceso a dicho repositorio. Antes de eliminar una etiqueta de un repositorio, asegúrese de revisar cualquier política de IAM que pueda utilizar la clave o el valor de una etiqueta para controlar el acceso a recursos como los repositorios. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Eliminar una etiqueta de un repositorio (consola)

Puedes usar la CodeCommit consola para eliminar la asociación entre una etiqueta y un CodeCommit repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas etiquetas desea eliminar.
3. En el panel de navegación, seleccione Configuración. Elija Repository tags (Etiquetas de repositorio).
4. Elija Edit (Edición de).
5. Busque la etiqueta que desea eliminar y, a continuación, seleccione Remove tag (Eliminar etiqueta).
6. Cuando haya terminado de eliminar las etiquetas, seleccione Submit (Enviar).

Eliminar una etiqueta de un repositorio (AWS CLI)

Siga estos pasos para usar el AWS CLI para eliminar una etiqueta de un CodeCommit repositorio. Al quitar una etiqueta no la elimina, sino que simplemente elimina la asociación entre la etiqueta y el repositorio.

 Note

Si eliminas un CodeCommit repositorio, todas las asociaciones de etiquetas se eliminarán del repositorio eliminado. No es necesario quitar las etiquetas antes de eliminar un repositorio.

En el terminal o en la línea de comandos, ejecute el comando `untag-resource`, especificando el nombre de recurso de Amazon (ARN) del repositorio donde desea quitar etiquetas y la clave de la etiqueta que desea quitar. Por ejemplo, para eliminar una etiqueta de un repositorio denominado *MyDemoRepo* con la clave de etiqueta *Status*:

```
aws codecommit untag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tag-keys Status
```

Si se ejecuta correctamente, este comando no devuelve nada. Para verificar las etiquetas asociadas con el repositorio, ejecute el comando `list-tags-for-resource`.

Gestiona los desencadenantes de un AWS CodeCommit repositorio

Puede configurar un CodeCommit repositorio para que las inserciones de código u otros eventos desencadenen acciones, como enviar una notificación desde Amazon Simple Notification Service (Amazon SNS) o invocar una función en AWS Lambda. Puede crear hasta 10 activadores para cada repositorio. CodeCommit

Los disparadores se suelen configurar para:

- Enviar correos electrónicos a los usuarios suscritos cada vez que alguien realiza inserciones en el repositorio.
- Enviar notificaciones a un sistema externo para que inicie una compilación después de que se realice una inserción en la ramificación principal del repositorio.

Las situaciones como la notificación a un sistema de compilación externo requieren escribir una función de Lambda para interactuar con otras aplicaciones. Las situaciones relacionadas con correo electrónico solo requiere la creación de un tema de Amazon SNS.

En este tema se muestra cómo configurar los permisos que permiten CodeCommit activar acciones en Amazon SNS y Lambda. También incluye enlaces a ejemplos para crear, editar, probar y eliminar disparadores.

Temas

- [Cree el recurso y añada permisos para CodeCommit](#)
- [Ejemplo: crear un AWS CodeCommit desencadenante de un tema de Amazon SNS](#)
- [Ejemplo: crear un AWS CodeCommit disparador para un AWS Lambda función](#)
- [Ejemplo: crea un disparador en AWS CodeCommit para un existente AWS Lambda función](#)
- [Edite los activadores de un AWS CodeCommit repositorio](#)
- [Pruebe los desencadenantes de un AWS CodeCommit repositorio](#)
- [Elimine los activadores de un AWS CodeCommit repositorio](#)

Cree el recurso y añada permisos para CodeCommit

Puede integrar los temas de Amazon SNS y las funciones de Lambda con los activadores CodeCommit, pero primero debe crear y, a continuación, configurar los recursos con una política que conceda CodeCommit los permisos para interactuar con esos recursos. Debe crear el recurso en el mismo lugar que Región de AWS el CodeCommit repositorio. Por ejemplo, si el repositorio está en este de EE. UU. (Ohio) (us-east-2), el tema de Amazon SNS o la función de Lambda deben estar en este de EE. UU. (Ohio).

- Para los temas de Amazon SNS, no es necesario configurar políticas o permisos de IAM adicionales si el tema de Amazon SNS se crea con la misma cuenta que el repositorio. CodeCommit Puedes crear el CodeCommit activador en cuanto hayas creado el tema Amazon SNS y te hayas suscrito a él.
- Para obtener más información sobre la creación de temas en Amazon SNS, consulte la [documentación de Amazon SNS](#).
- Para obtener información sobre cómo utilizar Amazon SNS para enviar mensajes a las colas de Amazon SQS, consulte [Envío de mensajes a colas de Amazon SQS](#) en la Guía para desarrolladores de Amazon SNS.
- Para obtener información sobre cómo usar Amazon SNS para invocar una función de Lambda, consulte [Invocación de funciones de Lambda](#) en la Guía para desarrolladores de Amazon SNS.
- Puede configurar las funciones de Lambda mediante la creación del disparador en la consola de Lambda como parte de la función. Este es el método más sencillo, ya que los activadores creados

en la consola Lambda incluyen automáticamente los permisos necesarios para CodeCommit invocar la función Lambda. Si crea el activador en CodeCommit, debe incluir una política que permita CodeCommit invocar la función. Para obtener más información, consulte [Crear un disparador para una función de Lambda existente](#).

Ejemplo: crear un AWS CodeCommit desencadenante de un tema de Amazon SNS

Puede crear un activador para un CodeCommit repositorio de modo que los eventos de ese repositorio activen notificaciones desde un tema del Amazon Simple Notification Service (Amazon SNS). Es posible que le interese crear un disparador para un tema de Amazon SNS que permita a los usuarios suscribirse y recibir notificaciones sobre los eventos del repositorio como, por ejemplo, la eliminación de ramificaciones. También puede aprovechar la integración de los temas de Amazon SNS con otros servicios, como Amazon Simple Queue Service (Amazon SQS) y AWS Lambda

Note

- Debe apuntar el disparador a un tema de Amazon SNS existente, que es la acción que se realiza en respuesta a los eventos del repositorio. Para obtener información sobre cómo crear temas de Amazon SNS y suscribirse a ellos, consulte [Introducción a Amazon Simple Notification Service](#).
- Los temas FIFO (primero en entrar, primero en salir) de Amazon SNS no se admiten como activadores. CodeCommit

Temas

- [Crear un activador para un tema de Amazon SNS para un CodeCommit repositorio \(consola\)](#)
- [Crear un activador para un tema de Amazon SNS para un CodeCommit repositorio \(AWS CLI\)](#)

Crear un activador para un tema de Amazon SNS para un CodeCommit repositorio (consola)

1. Abra la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositories (Repositorios), seleccione aquel en el que quiera crear los disparadores para los eventos del repositorio.

3. En el panel de navegación del repositorio, elija Settings (Configuración) y, a continuación, seleccione Triggers (Disparadores).
4. Elija Create trigger (Crear disparador) y, a continuación, haga lo siguiente:
 - En Nombre del disparador, introduzca un nombre para el disparador (por ejemplo, *MyFirstTrigger*).
 - En Eventos, seleccione los eventos del repositorio que activan el envío de notificaciones por parte del tema de Amazon SNS.

Si elige All repository events, no podrá seleccionar ningún otro evento. Para elegir un subconjunto de eventos, elimine All repository events y, a continuación, elija uno o más eventos de la lista. Por ejemplo, si desea que el disparador solo se active cuando un usuario crea una ramificación o una etiqueta en el repositorio de CodeCommit, elimine All repository events (Todos los eventos del repositorio) y elija Create branch or tag (Crear bifurcación o etiqueta).

- Si desea que el disparador se aplique a todas las ramificaciones del repositorio, en Branches (Ramificaciones), deje la selección en blanco, ya que esta opción predeterminada se aplica el disparador para que todas las ramificaciones de forma automática. Si desea que este disparador se aplique a ramificaciones específicas solo, elija hasta 10 nombres de ramificaciones de la lista de ramificaciones del repositorio.
- En Elija el servicio que utilizar, seleccione Amazon SNS.
- En Amazon SNS, elija un nombre de tema de la lista o introduzca el ARN para el tema.

 Note

Los temas FIFO (primero en entrar, primero en salir) de Amazon SNS no se admiten como activadores. CodeCommit Debe elegir un tema de Amazon SNS cuyo tipo esté establecido en Estándar.

- En Datos personalizados, también puede introducir cualquier información opcional que desee incluir en la notificación que envía el tema de Amazon SNS (por ejemplo, el nombre del canal IRC que usan los desarrolladores para hablar del desarrollo en este repositorio). Este campo es una cadena. No se puede usar para transmitir parámetros dinámicos.
5. (Opcional) Elija Test trigger (Probar disparador). Este paso le ayuda a confirmar que ha configurado correctamente el acceso entre CodeCommit y el tema Amazon SNS. Utiliza el

tema de Amazon SNS para enviar una notificación de prueba con los datos del repositorio, si están disponibles. Si no se dispone de datos reales, la notificación de prueba contiene datos de ejemplo.

6. Seleccione **Create trigger** (Crear disparador) para acabar de crear el disparador.

Crear un activador para un tema de Amazon SNS para un CodeCommit repositorio (AWS CLI)

También puedes usar la línea de comandos para crear un activador para un tema de Amazon SNS en respuesta a eventos del CodeCommit repositorio, como cuando alguien envía una confirmación a tu repositorio.

Para crear un disparador para un tema de Amazon SNS

1. Abrir un editor de texto sin formato y cree un archivo JSON que especifique:

- Nombre del tema de Amazon SNS.

Note

Los temas FIFO (primero en entrar, primero en salir) de Amazon SNS no se admiten como activadores. CodeCommit Debe elegir un tema de Amazon SNS cuyo tipo esté establecido en Estándar.

- El repositorio y las ramificaciones que desee monitorizar con este disparador. (Si no especifica ramificaciones, el disparador se aplica a todas las ramificaciones del repositorio).
- Los eventos que activan este disparador.

Guarde el archivo.

Por ejemplo, para crear un activador para un repositorio denominado *MyDemoRepo* que publique todos los eventos del repositorio en un tema de Amazon SNS con el nombre *MySNSTopic* de dos ramas, *main* y: *preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
```

```

        "name": "MyFirstTrigger",
        "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
        "customData": "",
        "branches": [
            "main", "preprod"
        ],
        "events": [
            "all"
        ]
    }
]
}

```

Debe haber un bloque de disparadores en JSON para cada disparador en un repositorio. Para crear más de un disparador para el repositorio, incluya más de un bloque de disparadores en el JSON. Recuerde que todos los disparadores creados en este archivo son para el repositorio especificado. No puede crear disparadores de varios repositorios en un único archivo JSON. Por ejemplo, si desea crear dos disparadores para un repositorio, puede crear un archivo JSON con dos bloques de disparadores. En el siguiente ejemplo, no se especifican las ramificaciones del segundo disparador, por lo que ese disparador se aplica a todas las ramificaciones:

```

{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyFirstTrigger",
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
      "customData": "",
      "branches": [
        "main", "preprod"
      ],
      "events": [
        "all"
      ]
    },
    {
      "name": "MySecondTrigger",
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic2",
      "customData": "",
      "branches": [],
      "events": [

```

```
        "updateReference", "deleteReference"
    ]
  }
]
```

Puede crear disparadores para eventos que especifique, como por ejemplo cuando se inserta una confirmación en un repositorio. Entre los tipos de eventos se incluyen:

- `all`: todos los eventos del repositorio y las ramificaciones especificados.
- `updateReference`: si las confirmaciones se envían al repositorio y ramificaciones especificados.
- `createReference`: si se crea una nueva ramificación o etiqueta en el repositorio especificado.
- `deleteReference`: si se elimina una ramificación o etiqueta en el repositorio especificado.

 Note

Puede usar más de un tipo de evento en un disparador. Sin embargo, si especifica `all`, no podrá especificar ningún otro evento.

Para ver la lista completa de tipos de eventos válidos, en el terminal o el símbolo del sistema, introduzca `aws codecommit put-repository-triggers help`.

Además, puede incluir una cadena en `customData` (por ejemplo, el nombre del canal IRC que usan los desarrolladores para hablar del desarrollo en este repositorio). Este campo es una cadena. No se puede usar para transmitir parámetros dinámicos. Esta cadena se añade como atributo al CodeCommit JSON devuelto en respuesta al disparador.

2. (Opcional) Ejecute el comando `test-repository-triggers` en un terminal o la línea de comandos. Esta prueba utiliza datos de ejemplo del repositorio (o genera datos de ejemplo si no hay datos disponibles) para enviar una notificación a los suscriptores del tema de Amazon SNS. Por ejemplo, se utiliza lo siguiente para comprobar que el JSON del archivo activador denominado *trigger.json* es válido y se CodeCommit puede publicar en el tema Amazon SNS:

```
aws codecommit test-repository-triggers --cli-input-json file://trigger.json
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

3. En el terminal o el símbolo del sistema, ejecute el comando `put-repository-triggers` para crear el disparador en CodeCommit. Por ejemplo, para usar un archivo JSON llamado `trigger.json` para crear el disparador:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Este comando devuelve un [ID de configuración](#) similar a este:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

4. Para ver la configuración del disparador, ejecute el comando `get-repository-triggers` y especifique el nombre del repositorio:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Este comando devuelve la estructura de todos los disparadores configurados para el repositorio, de forma similar a la siguiente:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE",
  "triggers": [
    {
      "events": [
        "all"
      ],
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
      "branches": [
        "main",
        "preprod"
      ]
    }
  ]
}
```

```

    ],
    "name": "MyFirstTrigger",
    "customData": "Project ID 12345"
  }
]
}

```

5. Para probar la funcionalidad del disparador, cree y envíe una confirmación al repositorio en el que haya configurado el disparador. Debería ver una respuesta del tema de Amazon SNS. Por ejemplo, si configura el tema de Amazon SNS para que envíe un correo electrónico, debería ver un mensaje de correo electrónico de Amazon SNS en la cuenta de correo electrónico suscrita al tema.

A continuación se muestra un ejemplo del resultado de un correo electrónico enviado desde Amazon SNS en respuesta a un envío a un CodeCommit repositorio:

```

{
  "Records": [
    {
      "awsRegion": "us-east-2",
      "codecommit": {
        "references": [
          {
            "commit": "317f8570EXAMPLE",
            "created": true,
            "ref": "refs/heads/NewBranch"
          },
          {
            "commit": "4c925148EXAMPLE",
            "ref": "refs/heads/preprod",
          }
        ]
      },
      "eventId": "1111-EXAMPLE-ID",
      "eventName": "ReferenceChange",
      "eventPartNumber": 1,
      "eventSource": "aws:codecommit",
      "eventSourceARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "eventTime": "2016-02-09T00:08:11.743+0000",
      "eventTotalParts": 1,
      "eventTriggerConfigId": "0123456-I-AM-AN-EXAMPLE",
      "eventTriggerName": "MyFirstTrigger",
      "eventVersion": "1.0",
    }
  ]
}

```

```
    "customData": "Project ID 12345",
    "userIdentityARN": "arn:aws:iam::111122223333:user/JaneDoe-CodeCommit",
  }
]
```

Ejemplo: crear un AWS CodeCommit disparador para un AWS Lambda función

Puede crear un disparador para un CodeCommit repositorio de modo que los eventos del repositorio invoquen una función Lambda. En este ejemplo, se crea una función Lambda que devuelve la URL utilizada para clonar el repositorio en un registro de Amazon CloudWatch .

Temas

- [Crear la función de Lambda](#)
- [Vea el activador de la función Lambda en la AWS CodeCommit repositorio](#)

Crear la función de Lambda

Al utilizar la consola Lambda para crear la función, también puede crear un CodeCommit activador para la función Lambda. Los siguientes pasos incluyen una función de Lambda de ejemplo. El ejemplo está disponible en dos idiomas: JavaScript y Python. La función devuelve las direcciones URL utilizadas para clonar un repositorio en un registro. CloudWatch

Para crear una función de Lambda mediante un esquema de Lambda

1. Inicie sesión en Consola de administración de AWS y abra la AWS Lambda consola en. <https://console.aws.amazon.com/lambda/>
2. En la página Funciones de Lambda, seleccione Crear una función. (Si no ha utilizado Lambda antes, seleccione Comience ahora).
3. En la página Crear función, elija Diseñar desde cero. En Nombre de función, proporciona un nombre para la función, por ejemplo *MyLambdaFunctionforCodeCommit*. En Runtime (Motor de ejecución), elija el idioma que desea utilizar para escribir su función y, a continuación, elija Create function (Crear función).
4. En la pestaña Configuration (Configuración), elija Add trigger (Agregar desencadenador).

5. En la configuración del activador, elija una opción CodeCommit de la lista desplegable de servicios.

Lambda > Add trigger

Add trigger

Trigger configuration

 CodeCommit

aws developer-tools git

Repository name
Select the repository to add a trigger to.

MyDemoRepo

Trigger name
Provide a name for the trigger that will invoke this function.

MyLambdaFunctionTrigger

Events
Choose one or more events to listen for. If you choose "All repository events", you cannot choose other event types.

Push to existing branch

Branch names
This trigger will be configured for all repository branches and tags by default. For a more specific configuration, choose up to 10 branches. If you choose "All branches", you cannot choose specific branches.

All branches

Custom data - optional
Custom data is additional contextual information used to distinguish this trigger from other triggers that run for the same event, refer to external resources, or group triggers from different repositories. For example, you could include the channel ID # for a chat room used by your team to collaborate on development.

#

Lambda will add the necessary permissions for AWS CodeCommit to invoke your Lambda function from this trigger.
[Learn more](#) about the Lambda permissions model.

Cancel Add

- En Nombre del repositorio, elija el nombre del repositorio donde desea configurar el disparador que utiliza la función de Lambda en respuesta a los eventos del repositorio.
- En Nombre del disparador, introduzca un nombre para el disparador (por ejemplo, *MyLambdaFunctionTrigger*).
- En Eventos, seleccione los eventos del repositorio que activan la función de Lambda. Si elige All repository events, no podrá seleccionar ningún otro evento. Si desea seleccionar un subconjunto de eventos, borre All repository events y seleccione los eventos que quiera de la lista. Por ejemplo, si desea que el activador se ejecute solo cuando un usuario cree una etiqueta o una rama en el AWS CodeCommit repositorio, elimine Todos los eventos del repositorio y, a continuación, elija Crear rama o etiqueta.
- Si desea que el disparador se aplique a todas las ramificaciones del repositorio, en Branches, elija All branches. En caso contrario, elija Specific branches. La ramificación predeterminada para el repositorio se añade por defecto. Puede conservar o eliminar esta ramificación de la lista. Seleccione hasta un máximo de diez nombres de ramificaciones de la lista de ramificaciones del repositorio.
- (Opcional) En Datos personalizados, introduzca la información que desee incluir en la función de Lambda (por ejemplo, el nombre del canal IRC que usan los desarrolladores para hablar del desarrollo del repositorio). Este campo es una cadena. No se puede usar para transmitir parámetros dinámicos.

Elija Añadir.

6. En la página Configuration (Configuración), en Function code (Código de función), en el tipo de entrada de código, elija Edit code inline (Editar código insertado). En Runtime, selecciona Node.js. Si desea crear una función de ejemplo en Python, elija Python.
7. En Code entry type, elija Edit code inline y, a continuación, sustituya el código "hello world" por uno de los dos siguientes ejemplos.

Para Node.js:

```
import {
    CodeCommitClient,
    GetRepositoryCommand,
} from "@aws-sdk/client-codecommit";

const codecommit = new CodeCommitClient({ region: "your-region" });
```

```

/**
 * @param {{ Records: { codecommit: { references: { ref: string }[] },
 * eventSourceARN: string }[] } event
 */
export const handler = async (event) => {
  // Log the updated references from the event
  const references = event.Records[0].codecommit.references.map(
    (reference) => reference.ref,
  );
  console.log("References:", references);

  // Get the repository from the event and show its git clone URL
  const repository = event.Records[0].eventSourceARN.split(":")[5];
  const params = {
    repositoryName: repository,
  };

  try {
    const data = await codecommit.send(new GetRepositoryCommand(params));
    console.log("Clone URL:", data.repositoryMetadata.cloneUrlHttp);
    return data.repositoryMetadata.cloneUrlHttp;
  } catch (error) {
    console.error("Error:", error);
    throw new Error(
      `Error getting repository metadata for repository ${repository}`,
    );
  }
};

```

Para Python:

```

import json
import boto3

codecommit = boto3.client("codecommit")

def lambda_handler(event, context):
    # Log the updated references from the event
    references = {
        reference["ref"]
        for reference in event["Records"][0]["codecommit"]["references"]
    }

```

```
}
print("References: " + str(references))

# Get the repository from the event and show its git clone URL
repository = event["Records"][0]["eventSourceARN"].split(":")[5]
try:
    response = codecommit.get_repository(repositoryName=repository)
    print("Clone URL: " + response["repositoryMetadata"]["cloneUrlHttp"])
    return response["repositoryMetadata"]["cloneUrlHttp"]
except Exception as e:
    print(e)
    print(
        "Error getting repository {}. Make sure it exists and that your
repository is in the same region as this function.".format(
            repository
        )
    )
    raise e
```

8. En la pestaña Permisos, en Rol de ejecución, elija el rol para abrirlo en la consola de IAM. Edite la política asociada para agregar el permiso `GetRepository` del repositorio en el que desea utilizar el desencadenador.

Vea el activador de la función Lambda en la AWS CodeCommit repositorio

Una vez que haya creado la función de Lambda, puede ver y probar el disparador en AWS CodeCommit. Al probar el disparador, se ejecuta la función en respuesta a los eventos del repositorio que especifique.

Para ver y probar el disparador de la función de Lambda

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositories (Repositorios), elija el repositorio donde desea ver los disparadores.
3. En el panel de navegación del repositorio, elija Settings (Configuración) y, a continuación, seleccione Triggers (Disparadores).
4. Revise la lista de disparadores para el repositorio. Debería ver el disparador que ha creado en la consola de Lambda. Elíjala de la lista y, a continuación, elija Test trigger (Probar disparador). Esta opción intenta invocar la función con datos de muestra sobre el repositorio, incluido el ID de

confirmación más reciente para el repositorio. (Si no existe ningún historial de confirmaciones, se generan valores de muestra compuestos de ceros). Esto le ayuda a confirmar que ha configurado correctamente el acceso entre AWS CodeCommit y la función Lambda.

5. Para continuar comprobando la funcionalidad del disparador, cree y envíe una confirmación al repositorio en el que haya configurado el disparador. Debería ver una respuesta de la función Lambda en la pestaña Monitorización para esa función de Lambda en la consola. En la pestaña Supervisión, seleccione Ver los inicios de sesión CloudWatch. La CloudWatch consola se abre en una pestaña nueva y muestra los eventos de su función. Seleccione el flujo de registro de la lista correspondiente al momento en el que envió la confirmación. Debería ver información del evento similar a esta:

```
START RequestId: 70afdc9a-EXAMPLE Version: $LATEST
2015-11-10T18:18:28.689Z 70afdc9a-EXAMPLE References: [ 'refs/heads/main' ]
2015-11-10T18:18:29.814Z 70afdc9a-EXAMPLE Clone URL: https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
END RequestId: 70afdc9a-EXAMPLE
REPORT RequestId: 70afdc9a-EXAMPLE Duration: 1126.87 ms Billed Duration: 1200 ms
Memory Size: 128 MB Max Memory Used: 14 MB
```

Ejemplo: crea un disparador en AWS CodeCommit para un existente AWS Lambda función

La forma más sencilla de crear un disparador que invoque una función de Lambda es crear ese disparador en la consola de Lambda. Esta integración integrada garantiza que CodeCommit cuente con los permisos necesarios para ejecutar la función. Para añadir un disparador para una función de Lambda existente, vaya a la consola de Lambda y elija la función. En la pestaña Triggers (Disparadores) de la función, siga los pasos que se indican en Add trigger (Añadir disparador). Estos pasos son similares a los que se muestran en [Crear la función de Lambda](#).

También puede crear un activador para una función Lambda en un CodeCommit repositorio. Para hacerlo, tendrá que seleccionar una función de Lambda existente que se va a invocar. También requiere que configure manualmente los permisos necesarios CodeCommit para ejecutar la función.

Temas

- [Configurar manualmente los permisos para CodeCommit permitir la ejecución de una función Lambda](#)
- [Crear un disparador para la función Lambda en un CodeCommit repositorio \(consola\)](#)

- [Crear un activador para una función Lambda para un CodeCommit repositorio \(AWS CLI\)](#)

Configurar manualmente los permisos para CodeCommit permitir la ejecución de una función Lambda

Si crea un disparador CodeCommit que invoca una función Lambda, debe configurar manualmente los permisos que CodeCommit permiten ejecutar la función Lambda. Para evitar la configuración manual, considere la posibilidad de crear el disparador para la función en la consola de Lambda.

Para CodeCommit permitir la ejecución de una función Lambda

1. Abra un editor de texto sin formato y cree un archivo JSON que especifique el nombre de la función de Lambda, los detalles del CodeCommit repositorio y las acciones que desea permitir en Lambda, similar a lo siguiente:

```
{
  "FunctionName": "MyCodeCommitFunction",
  "StatementId": "1",
  "Action": "lambda:InvokeFunction",
  "Principal": "codecommit.amazonaws.com",
  "SourceArn": "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo",
  "SourceAccount": "111122223333"
}
```

2. Guarde el archivo como un archivo JSON con un nombre que le resulte fácil de recordar (por ejemplo, `.json`). *AllowAccessfromMyDemoRepo*
3. Usando el archivo JSON que acaba de crear, en el terminal (Linux, macOS o Unix) o la línea de comandos (Windows), ejecute el comando `aws lambda add-permissions` para añadir un permiso a la política de recursos asociada con su función de Lambda:

```
aws lambda add-permission --cli-input-json file://AllowAccessfromMyDemoRepo.json
```

Este comando devuelve el JSON de la instrucción de directiva que acaba de añadir, de manera similar a esta:

```
{
  "Statement": "{\"Condition\":{\"StringEquals\":{\"AWS:SourceAccount\": \"111122223333\"}, \"ArnLike\":{\"AWS:SourceArn\": \"arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo\"}}}, \"Action
```

```
\":[\"lambda:InvokeFunction\"],\"Resource\": \"arn:aws:lambda:us-east-1:111122223333:function:MyCodeCommitFunction\", \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"codecommit.amazonaws.com\"}, \"Sid\": \"1\"}
}
```

Para obtener más información sobre las políticas de recursos para las funciones Lambda, consulte [AddPermissionThe Pull/Push Event Models](#) en la Guía del AWS Lambda usuario.

4. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
5. En el panel de navegación del panel de control, elija Funciones y, en la lista de funciones, seleccione *lambda_basic_execution*.
6. En la página de resumen del rol, elija la pestaña Permissions (Permisos) y en Inline Policies (Políticas en línea), seleccione Create Role Policy (Crear política de rol).
7. En la página Set Permissions, elija Policy Generator y después elija Select.
8. En la página Edit Permissions, lleve a cabo alguna de las siguientes operaciones:
 - En Effect, elija Allow.
 - En Servicio de AWS , seleccione AWS CodeCommit.
 - En Acciones, seleccione GetRepository.
 - En Amazon Resource Name (ARN), introduzca el ARN para el repositorio (por ejemplo, `arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo`).

Elija Add statement (Añadir declaración) y, luego, seleccione Next Step (Paso siguiente).

9. En la página Review Policy (Revisar política), elija Apply Policy (Aplicar política).

Su instrucción de directiva debe parecerse a los siguientes ejemplos:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt11111111",
      "Effect": "Allow",
      "Action": [
        "codecommit:GetRepository"
      ]
    }
  ]
}
```

```
    ],
    "Resource": [
      "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo"
    ]
  }
}
```

Crear un disparador para la función Lambda en un CodeCommit repositorio (consola)

Una vez creada la función Lambda, puede crear un disparador CodeCommit que ejecute la función en respuesta a los eventos del repositorio que especifique.

Note

Antes de poder probar o ejecutar correctamente el desencadenador del ejemplo, debe configurar las políticas que permiten CodeCommit invocar la función y la función Lambda para obtener información sobre el repositorio. Para obtener más información, consulte [Para CodeCommit permitir la ejecución de una función Lambda](#).

Para crear un disparador para una función de Lambda

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositories (Repositorios), seleccione aquel en el que quiera crear los disparadores para los eventos del repositorio.
3. En el panel de navegación del repositorio, elija Settings (Configuración) y, a continuación, seleccione Triggers (Disparadores).
4. Elija Create trigger.
5. En Create trigger (Crear disparador), haga lo siguiente:
 - En Nombre del disparador, introduzca un nombre para el disparador (por ejemplo, *MyLambdaFunctionTrigger*).
 - En Eventos, seleccione los eventos del repositorio que activan la función de Lambda.

Si elige All repository events, no podrá seleccionar ningún otro evento. Si desea seleccionar un subconjunto de eventos, borre All repository events y seleccione los eventos que quiera

- de la lista. Por ejemplo, si desea que el disparador solo se active cuando un usuario crea una etiqueta o una ramificación en el repositorio de CodeCommit, elimine All repository events (Todos los eventos del repositorio) y elija Create branch or tag (Crear bifurcación o etiqueta).
- Si desea que el disparador se aplique a todas las ramificaciones del repositorio, en Branches (Ramificaciones), deje la selección en blanco, ya que esta opción predeterminada se aplica el disparador para que todas las ramificaciones de forma automática. Si desea que este disparador se aplique a ramificaciones específicas solo, elija hasta 10 nombres de ramificaciones de la lista de ramificaciones del repositorio.
 - En Choose the service to use (Elija el servicio que utilizar), seleccione AWS Lambda.
 - En función de Lambda, seleccione el nombre de la función de la lista o escriba el ARN de la función.
 - (Opcional) En Datos personalizados, introduzca la información que desee incluir en la función de Lambda (por ejemplo, el nombre del canal IRC que usan los desarrolladores para hablar del desarrollo del repositorio). Este campo es una cadena. No se puede usar para transmitir parámetros dinámicos.
6. (Opcional) Elija Test trigger (Probar disparador). Esta opción intenta invocar la función con datos de muestra sobre el repositorio, incluido el ID de confirmación más reciente para el repositorio. (Si no existe ningún historial de confirmaciones, se generan valores de muestra compuestos de ceros). Esto le ayuda a confirmar que ha configurado correctamente el acceso entre CodeCommit y la función Lambda.
 7. Seleccione Create trigger (Crear disparador) para acabar de crear el disparador.
 8. Para verificar la funcionalidad del disparador, cree y envíe una confirmación al repositorio en el que haya configurado el disparador. Debería ver una respuesta de la función Lambda en la pestaña Monitorización para esa función de Lambda en la consola.

Crear un activador para una función Lambda para un CodeCommit repositorio (AWS CLI)

También puede usar la línea de comandos para crear un activador para una función Lambda en respuesta a eventos del CodeCommit repositorio, como cuando alguien envía una confirmación a su repositorio.

Para crear un disparador para una función de Lambda

1. Abrir un editor de texto sin formato y cree un archivo JSON que especifique:

- El nombre de la función de Lambda.
- El repositorio y las ramificaciones que desee monitorizar con este disparador. (Si no especifica ramificaciones, el disparador se aplica a todas las ramificaciones del repositorio).
- Los eventos que activan este disparador.

Guarde el archivo.

Por ejemplo, si desea crear un activador para un repositorio denominado *MyDemoRepo* que publique todos los eventos del repositorio en una función Lambda denominada *MyCodeCommitFunction* para dos ramas, *main* y *preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyLambdaFunctionTrigger",
      "destinationArn": "arn:aws:lambda:us-
east-1:111122223333:function:MyCodeCommitFunction",
      "customData": "",
      "branches": [
        "main", "preprod"
      ],
      "events": [
        "all"
      ]
    }
  ]
}
```

Debe haber un bloque de disparadores en JSON para cada disparador en un repositorio. Para crear más de un disparador en un repositorio, incluya bloques adicionales en el JSON. Recuerde que todos los disparadores creados en este archivo son para el repositorio especificado. No puede crear disparadores de varios repositorios en un único archivo JSON. Por ejemplo, si desea crear dos disparadores para un repositorio, puede crear un archivo JSON con dos bloques de disparadores. En el siguiente ejemplo, no se especifican ramificaciones en el segundo bloque de disparadores, por lo que el disparador se aplica a todas las ramificaciones:

```
{
```

```

    "repositoryName": "MyDemoRepo",
    "triggers": [
      {
        "name": "MyLambdaFunctionTrigger",
        "destinationArn": "arn:aws:lambda:us-
east-1:111122223333:function:MyCodeCommitFunction",
        "customData": "",
        "branches": [
          "main", "preprod"
        ],
        "events": [
          "all"
        ]
      },
      {
        "name": "MyOtherLambdaFunctionTrigger",
        "destinationArn": "arn:aws:lambda:us-
east-1:111122223333:function:MyOtherCodeCommitFunction",
        "customData": "",
        "branches": [],
        "events": [
          "updateReference", "deleteReference"
        ]
      }
    ]
  }
}

```

Puede crear disparadores para eventos que especifique, como por ejemplo cuando se inserta una confirmación en un repositorio. Entre los tipos de eventos se incluyen:

- `all`: todos los eventos del repositorio y las ramificaciones especificados.
- `updateReference`: si las confirmaciones se envían al repositorio y ramificaciones especificados.
- `createReference`: si se crea una nueva ramificación o etiqueta en el repositorio especificado.
- `deleteReference`: si se elimina una ramificación o etiqueta en el repositorio especificado.

 Note

Puede usar más de un tipo de evento en un disparador. Sin embargo, si especifica `all`, no podrá especificar ningún otro evento.

Para ver la lista completa de tipos de eventos válidos, en el terminal o el símbolo del sistema, introduzca `aws codecommit put-repository-triggers help`.

Además, puede incluir una cadena en `customData` (por ejemplo, el nombre del canal IRC que usan los desarrolladores para hablar del desarrollo en este repositorio). Este campo es una cadena. No se puede usar para transmitir parámetros dinámicos. Esta cadena se añade como atributo al CodeCommit JSON devuelto en respuesta al desencadenador.

2. (Opcional) Ejecute el comando `test-repository-triggers` en un terminal o la línea de comandos. Por ejemplo, se utiliza lo siguiente para comprobar que el archivo JSON denominado *trigger.json* es válido y CodeCommit puede activar la función Lambda. Esta prueba utiliza datos de muestra para disparar la función si no hay datos reales disponibles.

```
aws codecommit test-repository-triggers --cli-input-json file://trigger.json
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "successfulExecutions": [
    "MyLambdaFunctionTrigger"
  ],
  "failedExecutions": []
}
```

3. En el terminal o el símbolo del sistema, ejecute el comando `put-repository-triggers` para crear el disparador en CodeCommit. Por ejemplo, para usar un archivo JSON llamado *trigger.json* para crear el activador:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Este comando devuelve un ID de configuración similar a este:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

4. Para ver la configuración del disparador, ejecute el comando `get-repository-triggers` y especifique el nombre del repositorio:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Este comando devuelve la estructura de todos los disparadores configurados para el repositorio, de forma similar a la siguiente:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE",
  "triggers": [
    {
      "events": [
        "all"
      ],
      "destinationArn": "arn:aws:lambda:us-east-1:111122223333:MyCodeCommitFunction",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyLambdaFunctionTrigger",
      "customData": "Project ID 12345"
    }
  ]
}
```

5. Para comprobar la funcionalidad del disparador, cree y envíe una confirmación al repositorio en el que haya configurado el disparador. Debería ver una respuesta de la función Lambda en la pestaña Monitorización para esa función de Lambda en la consola.

Edite los activadores de un AWS CodeCommit repositorio

Puede editar los desencadenadores que se han creado para un CodeCommit repositorio. Puede cambiar los eventos y las ramificaciones del disparador, la acción que se realizará en respuesta al evento y otros valores.

Temas

- [Edición de un disparador de un repositorio \(consola\)](#)
- [Edite un activador para un repositorio \(AWS CLI\)](#)

Edición de un disparador de un repositorio (consola)

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione aquel en el que quiera editar un disparador para los eventos del repositorio.
3. En el panel de navegación del repositorio, elija Settings (Configuración) y, a continuación, seleccione Triggers (Disparadores).
4. En la lista de parámetros del repositorio, seleccione el disparador que desea editar y, a continuación, elija Edit (Editar).
5. Realice los cambios que desee en el disparador y, a continuación, elija Save (Guardar).

Edite un activador para un repositorio (AWS CLI)

1. En un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows), ejecute el comando `get-repository-triggers` para crear un archivo JSON con la estructura de todos los disparadores configurados en el repositorio. Por ejemplo, para crear un archivo JSON denominado *MyTriggers.json* con la estructura de todos los activadores configurados para un repositorio denominado *MyDemoRepo*:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>MyTriggers.json
```

Este comando no devuelve nada, pero *MyTriggers.json* se crea un archivo con el nombre en el directorio en el que se ejecutó el comando.

2. Edite el archivo JSON en un editor de texto sin formato y realice los cambios en el bloque de activación del disparador que desea editar. Sustituya el par `configurationId` por un par `repositoryName`. Guarde el archivo.

Por ejemplo, si desea editar un activador cuyo nombre aparece *MyFirstTrigger* en el repositorio indicado para *MyDemoRepo* que se aplique a todas las ramas, `configurationId` `repositoryName` sustitúyalo por las preprod ramas `main` y quítelas por las especificadas *red*

italic text. De forma predeterminada, si no se especifican las ramificaciones, el disparador se aplica a todas las ramificaciones del repositorio:

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    }
  ]
}
```

3. En el terminal o la línea de comandos, ejecute el comando `put-repository-triggers`. Esto actualiza todos los activadores del repositorio, incluidos los cambios que haya realizado en el *MyFirstTrigger* activador:

```
aws codecommit put-repository-triggers --repository-name MyDemoRepo
file://MyTriggers.json
```

Este comando devuelve un ID de configuración similar a este:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

Pruebe los desencadenantes de un AWS CodeCommit repositorio

Puede probar los desencadenadores que se han creado para un CodeCommit repositorio. La prueba implica ejecutar el disparador con datos de muestra del repositorio, incluido el ID de confirmación más reciente. Si no existe ningún historial de confirmación para el repositorio, se generan valores

de muestra compuestos por ceros. Probar los activadores le ayuda a confirmar que ha configurado correctamente el acceso entre CodeCommit y el destino del disparador, ya sea una AWS Lambda función o una notificación de Amazon Simple Notification Service.

Temas

- [Prueba de un disparador para un repositorio \(consola\)](#)
- [Pruebe un activador para un repositorio \(AWS CLI\)](#)

Prueba de un disparador para un repositorio (consola)

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione aquel en el que quiera probar un disparador para los eventos del repositorio.
3. En el panel de navegación del repositorio, elija Settings (Configuración) y, a continuación, seleccione Triggers (Disparadores).
4. Elija el disparador que desea probar y, a continuación, elija Test trigger (Probar disparador). Debería ver un mensaje sobre si el proceso se ha completado correctamente. Si es así, también debe ver una respuesta de acción correspondiente de la función de Lambda o el tema de Amazon SNS.

Pruebe un activador para un repositorio (AWS CLI)

1. En un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows), ejecute el comando `get-repository-triggers` para crear un archivo JSON con la estructura de todos los disparadores configurados en el repositorio. Por ejemplo, para crear un archivo JSON denominado *TestTrigger.json* con la estructura de todos los activadores configurados para un repositorio denominado MyDemoRepo:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>TestTrigger.json
```

Este comando crea un archivo cuyo nombre aparece *TestTriggers.json* en el directorio en el que se ejecutó el comando.

2. Edite el archivo JSON en un editor de texto sin formato y realice los cambios en la instrucción de disparo. Sustituya el par `configurationId` por un par `repositoryName`. Guarde el archivo.

Por ejemplo, si quiere probar un activador denominado *MyFirstTrigger* en el repositorio denominado para *MyDemoRepo* que se aplique a todas las ramas, sustitúyalo por un archivo `configurationId` con un aspecto similar al siguiente `repositoryName` y guárdelo como *TestTrigger.json*:

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    }
  ]
}
```

3. En el terminal o la línea de comandos, ejecute el comando `test-repository-triggers`. De este modo, se actualizan todos los activadores del repositorio, incluidos los cambios que hayas realizado en el *MyFirstTrigger* activador:

```
aws codecommit test-repository-triggers --cli-input-json file://TestTrigger.json
```

Este comando devuelve una respuesta similar a la siguiente:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```


Elimine los activadores de un AWS CodeCommit repositorio

Es posible que le interese eliminar disparadores si ya no los está usando. No puede deshacer la eliminación de un disparador, pero puede crearlos de nuevo.

Note

Si ha configurado uno o más disparadores en el repositorio, al eliminar el repositorio no se eliminarán los temas de Amazon SNS ni las funciones de Lambda que configuró como objetivos de dichos disparadores. Asegúrese de eliminar también esos recursos si ya no los necesita.

Temas

- [Eliminar un disparador de un repositorio \(consola\)](#)
- [Eliminar un disparador de un repositorio \(AWS CLI\)](#)

Eliminar un disparador de un repositorio (consola)

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione aquel en el que quiera eliminar los disparadores para los eventos del repositorio.
3. En el panel de navegación del repositorio, elija Settings. En Settings, elija Triggers.
4. Seleccione el disparador que desee eliminar de la lista de disparadores y, a continuación, seleccione Delete (Eliminar).
5. En el cuadro de diálogo, escriba delete (eliminar) para confirmar.

Eliminar un disparador de un repositorio (AWS CLI)

1. En un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows), ejecute el comando `get-repository-triggers` para crear un archivo JSON con la estructura de todos los disparadores configurados en el repositorio. Por ejemplo, para crear un archivo JSON denominado *MyTriggers.json* con la estructura de todos los activadores configurados para un repositorio denominado MyDemoRepo:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
>MyTriggers.json
```

Este comando crea un archivo cuyo nombre aparece *MyTriggers.json* en el directorio en el que se ejecutó el comando.

2. Edite el archivo JSON en un editor de texto sin formato y elimine el bloque de disparo para el disparador que desee eliminar. Sustituya el par `configurationId` por un par `repositoryName`. Guarde el archivo.

Por ejemplo, si desea eliminar un desencadenador denominado *MyFirstTrigger* del repositorio denominado *MyDemoRepo*, debe `configurationId` `repositoryName` sustituirlo por y eliminar la sentencia *dered italic text*:

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-
east-2:111122223333:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    },
    {
      "destinationArn": "arn:aws:lambda:us-
east-2:111122223333:function:MyCodeCommitJSFunction",
      "branches": [],
      "name": "MyLambdaTrigger",
      "events": [
        "all"
      ]
    }
  ]
}
```

3. En el terminal o la línea de comandos, ejecute el comando `put-repository-triggers`. De este modo, se actualizan los activadores del repositorio y se elimina el *MyFirstTrigger* activador:

```
aws codecommit put-repository-triggers --repository-name MyDemoRepo  
file:///MyTriggers.json
```

Este comando devuelve un ID de configuración similar a este:

```
{  
  "configurationId": "0123456-I-AM-AN-EXAMPLE"  
}
```

Note

Para eliminar todos los disparadores de un repositorio denominado *MyDemoRepo*, su archivo JSON tendría un aspecto similar a este:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": []  
}
```

Amazon CodeGuru Reviewer y AWS CodeCommit

Important

A partir del 7 de noviembre de 2025, Amazon CodeGuru Reviewer se encuentra en modo de mantenimiento. No puede crear nuevas asociaciones de repositorios con CodeGuru Reviewer. Las asociaciones existentes seguirán funcionando, pero no AWS mejorarán ni añadirán funcionalidad al servicio.

Si ya tienes una asociación de CodeGuru revisores con un CodeCommit repositorio, este seguirá analizando las solicitudes de incorporación de datos como hasta ahora. Para desasociar un repositorio de CodeGuru Reviewer, selecciona Configuración, Amazon CodeGuru Reviewer y, a continuación, selecciona Desasociar repositorio.

Para obtener funciones de revisión de código y escaneo de seguridad, considere las siguientes alternativas:

- Amazon Q Developer: proporciona una revisión del código que incluye análisis estático, detección de secretos y escaneo de dependencias. Funciona en su IDE GitHub y con GitLab los repositorios. Para obtener más información, consulta Cómo [revisar el código con Amazon Q Developer](#).
- Amazon Inspector: descubre y escanea automáticamente los repositorios de código en busca de vulnerabilidades de software. Para obtener más información, consulte [Amazon Inspector Code Security](#).

Para obtener más información sobre el cambio de disponibilidad de Amazon CodeGuru Reviewer, consulte Cambio de [disponibilidad de Amazon CodeGuru Reviewer](#).

Vea los detalles CodeCommit del repositorio

Puedes usar la AWS CodeCommit consola o Git desde un repositorio local conectado al CodeCommit repositorio para ver información sobre los repositorios disponibles. AWS CLI

Antes de seguir estas instrucciones, realice los pasos de [Configuración](#).

Temas

- [Visualización de la información de los repositorios \(consola\)](#)
- [Ver detalles CodeCommit del repositorio \(Git\)](#)
- [Ver los detalles del CodeCommit repositorio \(AWS CLI\)](#)

Visualización de la información de los repositorios (consola)

Utilice la AWS CodeCommit consola para ver rápidamente todos los repositorios creados con su cuenta de Amazon Web Services.

1. Abra la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositorios, consulte los detalles sobre los repositorios en la Región de AWS donde haya iniciado sesión. En el selector de regiones, seleccione una Región de AWS diferente para ver los repositorios de esa región.
3. Elija el nombre del repositorio del que desea ver más detalles y luego realice una de las siguientes acciones:

- Si desea ver la dirección URL para clonar el repositorio, en el panel de navegación, seleccione Clone URL (Clonar URL) y, a continuación, elija el protocolo que desea utilizar para clonar el repositorio. Con esto se copia la URL clonada. Para revisarla, péguela en un editor de texto sin formato.
- Para ver las opciones configurables del repositorio, así como detalles como el ARN y el ID del repositorio, en el panel de navegación, seleccione Configuración.

Note

Si ha iniciado sesión como un usuario de IAM, puede configurar y guardar sus preferencias para ver el código y otros ajustes de la consola. Para obtener más información, consulte [Trabajo con las preferencias del usuario](#).

Ver detalles CodeCommit del repositorio (Git)

Para usar Git desde un repositorio local para ver detalles sobre los CodeCommit repositorios, ejecuta el `git remote show` comando.

Antes de realizar estos pasos, conecta el repositorio local al repositorio. CodeCommit Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Ejecuta el `git remote show remote-name` comando, donde **remote-name** está el alias del CodeCommit repositorio (de forma predeterminada, `origin`).

Tip

Para obtener una lista de los nombres de los CodeCommit repositorios y sus direcciones URL, ejecute el `git remote -v` comando.

Por ejemplo, para ver los detalles del CodeCommit repositorio con el alias `origin`:

```
git remote show origin
```

2. Para HTTPS:

```
* remote origin
Fetch URL: https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
Push URL: https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
HEAD branch: (unknown)
Remote branches:
  MyNewBranch tracked
  main tracked
Local ref configured for 'git pull':
  MyNewBranch merges with remote MyNewBranch (up to date)
Local refs configured for 'git push':
  MyNewBranch pushes to MyNewBranch (up to date)
  main pushes to main (up to date)
```

Para SSH:

```
* remote origin
Fetch URL: ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
Push URL: ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
HEAD branch: (unknown)
Remote branches:
  MyNewBranch tracked
  main tracked
Local ref configured for 'git pull':
  MyNewBranch merges with remote MyNewBranch (up to date)
Local refs configured for 'git push':
  MyNewBranch pushes to MyNewBranch (up to date)
  main pushes to main (up to date)
```

 Tip

Para buscar el ID de clave de SSH del usuario de IAM, abra la consola de IAM y amplíe Credenciales de seguridad en la página de información del usuario de IAM. El ID de la clave SSH se encuentra en SSH Keys for. AWS CodeCommit

Para ver más opciones, consulte la documentación de Git.

Ver los detalles del CodeCommit repositorio (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar el AWS CLI para ver los detalles del repositorio, ejecute los siguientes comandos:

- Para ver una lista de los nombres de los CodeCommit repositorios y sus correspondientes ID, ejecuta [list-repositories](#).
- [Para ver información sobre un único CodeCommit repositorio, ejecute get-repository.](#)
- [Para ver información sobre varios repositorios, ejecute batch-get-repositories. CodeCommit](#)

Para ver una lista de repositorios CodeCommit

1. Ejecute el comando list-repositories:

```
aws codecommit list-repositories
```

También puede utilizar las opciones `--sort-by` o `--order` si desea cambiar el orden en el que se muestra la información devuelta.

2. Si se ejecuta correctamente, este comando genera un `repositories` objeto que contiene los nombres e ID de todos los repositorios CodeCommit asociados a la cuenta de Amazon Web Services.

A continuación, se muestra un ejemplo de salida basado en el comando anterior:

```
{
  "repositories": [
    {
      "repositoryName": "MyDemoRepo",
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
    },
    {
      "repositoryName": "MyOtherDemoRepo",
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
    }
  ]
}
```

Para ver los detalles de un único repositorio CodeCommit

1. Ejecute el `get-repository` comando y especifique el nombre del CodeCommit repositorio con la `--repository-name` opción.

Tip

Para obtener el nombre del CodeCommit repositorio, ejecute el comando [list-repositories](#).

Por ejemplo, para ver los detalles de un CodeCommit repositorio llamado: MyDemoRepo

```
aws codecommit get-repository --repository-name MyDemoRepo
```

2. Si la operación se completó correctamente, el comando devuelve un objeto `repositoryMetadata` con la siguiente información:
 - El nombre del repositorio (`repositoryName`).
 - La descripción del repositorio (`repositoryDescription`).
 - El ID exclusivo del repositorio generado por el sistema (`repositoryId`).
 - El ID y la cuenta de Amazon Web Services asociados con el repositorio (`accountId`).

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "repositoryMetadata": {
    "creationDate": 1429203623.625,
    "defaultBranch": "main",
    "repositoryName": "MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "lastModifiedDate": 1430783812.0869999,
    "repositoryDescription": "My demonstration repository",
    "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
    "accountId": "111111111111"
```



```
}  
}
```

Para ver los detalles de varios CodeCommit repositorios

1. Ejecute el comando `batch-get-repositories` con la opción `--repository-names`. Añada un espacio entre el nombre de cada CodeCommit repositorio.

Tip

Para obtener los nombres de los repositorios CodeCommit, ejecute el comando [list-repositories](#).

Por ejemplo, para ver los detalles de dos CodeCommit repositorios denominados y: `MyDemoRepo` `MyOtherDemoRepo`

```
aws codecommit batch-get-repositories --repository-names MyDemoRepo MyOtherDemoRepo
```

2. Si la operación se completó correctamente, el comando devuelve un objeto con la siguiente información:
 - Una lista de todos CodeCommit los repositorios que no se pudieron encontrar (`repositoriesNotFound`).
 - Una lista de CodeCommit repositorios (`repositories`). El nombre de cada CodeCommit repositorio va seguido de:
 - La descripción del repositorio (`repositoryDescription`).
 - El ID exclusivo del repositorio generado por el sistema (`repositoryId`).
 - El ID y la cuenta de Amazon Web Services asociados con el repositorio (`accountId`).

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{  
  "repositoriesNotFound": [],  
  "repositories": [  
    {  
      "creationDate": 1429203623.625,
```

```

        "defaultBranch": "main",
        "repositoryName": "MyDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/
repos/MyDemoRepo",
        "lastModifiedDate": 1430783812.0869999,
        "repositoryDescription": "My demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/
repos/MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
        "accountId": "111111111111"
    },
    {
        "creationDate": 1429203623.627,
        "defaultBranch": "main",
        "repositoryName": "MyOtherDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/
repos/MyOtherDemoRepo",
        "lastModifiedDate": 1430783812.0889999,
        "repositoryDescription": "My other demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/
repos/MyOtherDemoRepo",
        "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo",
        "accountId": "111111111111"
    }
],
"repositoriesNotFound": []
}

```

Cambio AWS CodeCommit configuración del repositorio

Puede utilizar la consola AWS CLI y la AWS CodeCommit consola para cambiar la configuración de un CodeCommit repositorio, como su descripción o su nombre.

Important

Pero tenga en cuenta que si cambia el nombre de un repositorio, podría invalidar cualquier repositorio local que utilice el nombre antiguo en su URL remota. Ejecute el comando

`git remote set-url` para actualizar la URL remota que va a utilizar el nuevo nombre del repositorio.

Temas

- [Cambio de la configuración de un repositorio \(consola\)](#)
- [Cambio AWS CodeCommit configuración del repositorio \(AWS CLI\)](#)

Cambio de la configuración de un repositorio (consola)

Para usar la AWS CodeCommit consola para cambiar la configuración de un CodeCommit repositorio AWS CodeCommit, sigue estos pasos.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuya configuración desea cambiar.
3. En el panel de navegación, seleccione Configuración.
4. Para cambiar el nombre del repositorio Repository name (Nombre del repositorio), introduzca un nuevo nombre en el cuadro de texto Name (Nombre) y elija Save (Guardar). Cuando se lo pidan, verifique su elección.

Important

Si se cambia el nombre del AWS CodeCommit repositorio, se cambiarán las URL SSH y HTTPS que los usuarios necesitan para conectarse al repositorio. Para que los usuarios puedan conectarse a este repositorio, tendrán que actualizar la configuración de la conexión. Además, puesto que el repositorio cambiará el ARN, al cambiar el nombre del repositorio, se invalidarán las políticas del usuario de IAM que confían en este ARN del repositorio.

Para conectarse al repositorio tras cambiar el nombre, los usuarios deberán ejecutar el comando `git remote set-url` y especificar la nueva URL que se va a utilizar. Por ejemplo, si cambias el nombre del repositorio de `MyDemoRepo` a `MyRenamedDemoRepo`, los usuarios que usen HTTPS para conectarse al repositorio ejecutarán el siguiente comando de Git:

```
git remote set-url origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Los usuarios que utilicen SSH para conectarse al repositorio deberán ejecutar el siguiente comando de Git:

```
git remote set-url origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Para ver más opciones, consulte la documentación de Git.

5. Para cambiar la descripción del repositorio, modifique el texto que aparece en el cuadro de texto Description (Descripción) y haga clic en Save (Guardar).

 Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si eres un desarrollador de aplicaciones que utiliza BatchGetRepositories las API GetRepository o las API y tienes pensado mostrar el campo de descripción del repositorio en un navegador web, consulta la [referencia de la CodeCommit API](#).

6. Para cambiar la ramificación predeterminada, en Default branch (Ramificación personalizada) elija una ramificación diferente en la lista desplegable. Seleccione Save.
7. Para cambiar la clave de AWS KMS cifrado utilizada para cifrar y descifrar los datos del repositorio, en Clave de cifrado del repositorio, elija una clave gestionada por el cliente Clave administrada de AWS o una clave gestionada por el cliente para especificar el tipo de clave que se va a utilizar. Si elige una clave administrada por el cliente, introduzca el ARN de la clave. Seleccione Save.
8. Para eliminar el repositorio, elija Delete repository. En el cuadro junto a Type the name of the repository to confirm deletion (Escriba el nombre del repositorio para confirmar la eliminación), introduzca **deletey**, a continuación, seleccione Delete (Eliminar).

 Important

Tras eliminar este repositorio AWS CodeCommit, ya no podrá clonarlo en ningún repositorio local o compartido. Tampoco podrá extraer ni enviar datos desde ningún repositorio local ni repositorio compartido. Esta acción no se puede deshacer.

Cambio AWS CodeCommit configuración del repositorio (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

AWS CLI Para cambiar la configuración de un CodeCommit repositorio AWS CodeCommit, ejecute uno o más de los siguientes comandos:

- [update-repository-description para cambiar la descripción](#) de un repositorio. CodeCommit
- [update-repository-name para cambiar el nombre de un repositorio](#). CodeCommit

Para cambiar la descripción de un repositorio CodeCommit

1. Ejecute el comando `update-repository-description`, especificando:

- El nombre del CodeCommit repositorio (con la `--repository-name` opción).

Tip

Para obtener el nombre del CodeCommit repositorio, ejecute el [list-repositories](#) comando.

- La descripción del nuevo repositorio (con la opción `--repository-description`).

Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si eres un desarrollador de aplicaciones que utiliza `BatchGetRepositories` las API `GetRepository` o las API y tienes pensado mostrar el campo de descripción del repositorio en un navegador web, consulta la [referencia de la CodeCommit API](#).

Por ejemplo, para cambiar la descripción del CodeCommit repositorio denominado `MyDemoRepo` a `This description was changed`:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --  
repository-description "This description was changed"
```

Este comando devuelve resultados solo si hay errores.

2. Para comprobar la descripción modificada, ejecute el `get-repository` comando y especifique el nombre del CodeCommit repositorio cuya descripción ha cambiado con la `--repository-name` opción.

El resultado del comando muestra el texto modificado en `repositoryDescription`.

Para cambiar el nombre de un CodeCommit repositorio

1. Ejecute el comando `update-repository-name`, especificando:
 - El nombre actual del CodeCommit repositorio (con la `--old-name` opción).

Tip

Para obtener el nombre del CodeCommit repositorio, ejecuta el comando [list-repositories](#).

- El nuevo nombre del CodeCommit repositorio (con la `--new-name` opción).

Por ejemplo, para cambiar el repositorio denominado `MyDemoRepo` a `MyRenamedDemoRepo`:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-name
MyRenamedDemoRepo
```

Este comando devuelve resultados solo si hay errores.

Important

Al cambiar el nombre del AWS CodeCommit repositorio, se modifican las URL SSH y HTTPS que los usuarios necesitan para conectarse al repositorio. Los usuarios no podrán conectarse a este repositorio hasta que actualicen la configuración de la conexión. Además, puesto que el repositorio cambia el ARN, al cambiar el nombre del repositorio, se invalidan las políticas del usuario de IAM que confían en este ARN del repositorio.

2. Para comprobar el nombre cambiado, ejecute el comando `list-repositories` y revise la lista de nombres de los repositorios.

Sincronice los cambios entre un repositorio local y un AWS CodeCommit repositorio

Usas Git para sincronizar los cambios entre un repositorio local y el CodeCommit repositorio conectado al repositorio local.

Para enviar los cambios del repositorio local al repositorio, ejecuta. CodeCommit `git push remote-name branch-name`

Para extraer los cambios del repositorio local desde el CodeCommit repositorio, ejecuta. `git pull remote-name branch-name`

Tanto para empujar como para tirar, **remote-name** es el apodo que el repositorio local usa para el CodeCommit repositorio. **branch-name** es el nombre de la rama del CodeCommit repositorio a la que se va a enviar o extraer.

Tip

Para obtener el apodo que el repositorio local usa para el CodeCommit repositorio, ejecute `git remote`. Para obtener una lista de los nombres de las ramificaciones, ejecute `git branch`. Un asterisco (*) aparecerá junto al nombre de la ramificación actual. (También puede ejecutar `git status` para mostrar el nombre de la ramificación actual).

Note

Si has clonado el repositorio, desde la perspectiva del repositorio local, no **remote-name** es el nombre del repositorio. CodeCommit Al clonar un repositorio, **remote-name** se establece automáticamente en. `origin`

Por ejemplo, para enviar los cambios del repositorio local a la `main` rama del CodeCommit repositorio con el apodo `origin`:

```
git push origin main
```

Del mismo modo, para extraer los cambios del repositorio local desde la main rama del CodeCommit repositorio con el apodo: origin

```
git pull origin main
```

Tip

Si añade la opción `-u` a `git push`, establece la información de seguimiento ascendente. Por ejemplo, si corres `git push -u origin main`, en el futuro podrás correr `git push` y `git pull` sin el *remote-namebranch-name*. Para obtener información de seguimiento ascendente, ejecute `git remote show remote-name` (por ejemplo, `git remote show origin`).

Para ver más opciones, consulte la documentación de Git.

Envío de confirmaciones a un repositorio de Git adicional

Puede configurar el repositorio local para enviar los cambios a dos repositorios remotos. Por ejemplo, puede que desee seguir utilizando la solución del repositorio de Git existente mientras prueba AWS CodeCommit. Sigue estos pasos básicos para enviar los cambios de tu repositorio local a un repositorio CodeCommit de Git independiente.

Tip

Si no tienes un repositorio de Git, puedes crear uno vacío en un servicio que no sea CodeCommit y, a continuación, migrar tu CodeCommit repositorio a él. Debe seguir unos pasos similares a los que se detallan en [Migrar a CodeCommit](#).

1. En el símbolo del sistema o el terminal, cambie al directorio del repositorio local y ejecute el comando `git remote -v`. Debería ver una salida similar a esta:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
```



```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para SSH:

```
origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

2. Ejecuta el `git remote set-url --add --push origin` ***git-repository-name*** comando where ***git-repository-name*** es la URL y el nombre del repositorio de Git donde quieres alojar tu código. Esto cambia el destino del envío de `origin` al repositorio de Git.

 Note

`git remote set-url --add --push` anula la URL predeterminada de los envíos, de modo que debe ejecutar este comando dos veces tal y como se indica más adelante.

Por ejemplo, el siguiente comando cambia la inserción de origen a ***some-URL/MyDestinationRepo***:

```
git remote set-url --add --push origin some-URL/MyDestinationRepo
```

Este comando no devuelve nada.

 Tip

Si estás accediendo a un repositorio de Git que requiere credenciales, asegúrate de configurarlas en un asistente de credenciales o en la configuración de la ***some-URL*** cadena. De lo contrario, los envíos a ese repositorio devolverán un error.

3. Si ejecuta el comando `git remote -v` de nuevo, debería ver un resultado similar a este:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin some-URL/MyDestinationRepo (push)
```

Para SSH:

```
origin  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  some-URL/MyDestinationRepo (push)
```

4. Ahora añada el CodeCommit repositorio. Ejecute `git remote set-url --add --push origin` de nuevo, esta vez con la URL y el nombre del repositorio de CodeCommit.

Por ejemplo, el siguiente comando añade el envío de origen a `https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo`:

Para HTTPS:

```
git remote set-url --add --push origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Para SSH:

```
git remote set-url --add --push origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Este comando no devuelve nada.

5. Si ejecuta el comando `git remote -v` de nuevo, debería ver un resultado similar a este:

Para HTTPS:

```
origin  https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  some-URL/MyDestinationRepo (push)
origin  https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para SSH:

```
origin  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  some-URL/MyDestinationRepo (push)
origin  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Ahora tienes dos repositorios de Git como destino para tus empujes, pero estos van primero a ***some-URL/***. MyDestinationRepo Si el envío a ese repositorio devuelve un error, las confirmaciones no se envían a ningún repositorio.

 Tip

Si el otro repositorio requiere credenciales que quieres introducir manualmente, considera la posibilidad de cambiar el orden de las inserciones para que las introduzcas primero. CodeCommit Ejecute el comando `git remote set-url --delete` para eliminar el repositorio al que se envían primero y, a continuación, ejecute `git remote set-url --add` para volver a añadirlo de modo que se convierta en el segundo destino de envío de la lista.

Para ver más opciones, consulte la documentación de Git.

6. Para comprobar que no se está enviado a los dos repositorios remotos, utilice un editor de texto para crear el siguiente archivo de texto en el repositorio local:

```
bees.txt
-----
Bees are flying insects closely related to wasps and ants, and are known for their
role in pollination and for producing honey and beeswax.
```

7. Ejecute `git add` para almacenar el cambio en el repositorio local:

```
git add bees.txt
```

8. Ejecute `git commit` para confirmar el cambio en el repositorio local:

```
git commit -m "Added bees.txt"
```

9. Para enviar la confirmación del repositorio local a tus repositorios remotos, ejecuta `git push -u remote-name branch-name` where, que ***remote-name*** es el apodo que el repositorio local usa para los repositorios remotos y ***branch-name*** es el nombre de la rama que se va a enviar al repositorio.

 Tip

Solo tiene que utilizar la opción `-u` la primera vez que realice un envío. A continuación, se establece la información de seguimiento ascendente.

Por ejemplo, la ejecución de `git push -u origin main` mostraría que el envío llegó a los repositorios remotos de las ramificaciones esperadas, con un resultado parecido a este:

Para HTTPS:

```
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To some-URL/MyDestinationRepo
    a5ba4ed..250f6c3  main -> main
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
    a5ba4ed..250f6c3  main -> main
```

Para SSH:

```
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To some-URL/MyDestinationRepo
    a5ba4ed..250f6c3  main -> main
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
```

```
Total 3 (delta 1), reused 0 (delta 0)
remote:
To ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
a5ba4ed..250f6c3  main -> main
```

Para ver más opciones, consulte la documentación de Git.

Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles

Puede configurar el acceso a los CodeCommit repositorios para los usuarios y grupos de IAM de otra cuenta. AWS Esto es lo que normalmente se conoce como acceso entre cuentas. En esta sección, se proporcionan ejemplos e instrucciones paso a paso para configurar el acceso multicuenta a un repositorio denominado *MySharedDemoRepo* en la región EE.UU. Este (Ohio) en una AWS cuenta (denominada AccountA) para los usuarios de IAM que pertenecen a un grupo de IAM *DevelopersWithCrossAccountRepositoryAccess* nombrado en AWS otra cuenta (denominado AccountB).

Esta sección se divide en tres partes:

- Acciones del administrador de CuentaA.
- Acciones del administrador de CuentaB.
- Acciones del usuario del repositorio de CuentaB.

Para configurar el acceso entre cuentas:

- El administrador de AccountA inicia sesión como usuario de IAM con los permisos necesarios para crear y gestionar repositorios y crear funciones en CodeCommit IAM. Si utilizas políticas gestionadas, aplica IAMFullAccess y AWSCodeCommitFullAccess a este usuario de IAM.

El ID de ejemplo de CuentaA es **111122223333**.

- El administrador de CuentaB inicia sesión como usuario de IAM con los permisos necesarios para crear y administrar usuarios y grupos de IAM, así como para configurar políticas para los usuarios y grupos. Si utiliza políticas administradas, aplíquelas IAMFullAccess a este usuario de IAM.

El ID de ejemplo de CuentaB es **888888888888**.

- El usuario del repositorio en AccountB, para emular las actividades de un desarrollador, inicia sesión como un usuario de IAM que es miembro del grupo de IAM creado para permitir el acceso al repositorio en AccountA. CodeCommit Esta cuenta debe configurarse con:
 - AWS Acceso a la consola de administración.
 - Una clave de acceso y una clave secreta para usar al conectarse a AWS los recursos y el ARN del rol que se debe asumir al acceder a los repositorios en AccountA.
 - La utilidad git-remote-codecommit en el equipo local donde se clona el repositorio. Esta utilidad requiere Python y su instalador, pip. Puede descargar la utilidad desde [git-remote-codecommit](#) en el sitio web de Python Package Index.

Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#) y [Usuarios de IAM](#).

Temas

- [Cross-account acceso al repositorio: acciones para el administrador en AccountA](#)
- [Cross-account acceso al repositorio: acciones para el administrador en AccountB](#)
- [Cross-account acceso al repositorio: acciones para el usuario del repositorio en AccountB](#)

Cross-account acceso al repositorio: acciones para el administrador en AccountA

Para permitir a los usuarios o grupos de Cuenta B obtener acceso a un repositorio en CuentaA, un administrador de CuentaA debe:

- Crear una política en CuentaA que conceda acceso al repositorio.
- Crear una función en CuentaA que puedan desempeñar los usuarios de IAM y grupos en Cuenta B.
- Asocie la política de al rol.

En las siguientes secciones se proporcionan pasos y ejemplos.

Temas

- [Paso 1: Crear una política de acceso al repositorio en CuentaA](#)
- [Paso 2: Crear una función para el acceso al repositorio en CuentaA](#)

Paso 1: Crear una política de acceso al repositorio en CuentaA

Puede crear una política en CuentaA que conceda acceso al repositorio de Cuenta A a los usuarios de CuentaB. Según el nivel de acceso que desee permitir, realice una de las siguientes acciones:

- Configure la política para permitir a los usuarios de Cuenta B obtener acceso a un repositorio específico, pero no les deje ver una lista de todos los repositorios en CuentaA.
- Configure acceso adicional para permitir a los usuarios de Cuenta B elegir el repositorio de una lista de todos los repositorios en CuentaA.

Para crear una política de acceso al repositorio

1. Inicie sesión en la consola AWS de administración como usuario de IAM con permisos para crear políticas en AccountA.
2. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
3. En el panel de navegación, seleccione Políticas.
4. Elija Crear política.
5. Seleccione la pestaña JSON y pegue el siguiente documento de política JSON en el cuadro de texto JSON. *us-east-2* Región de AWS Sustitúyalo por el del repositorio, *111122223333* por el ID de cuenta de AccountA y *MySharedDemoRepo* por el nombre de tu CodeCommit repositorio en AccountA:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGet*",
        "codecommit:Create*",
        "codecommit>DeleteBranch",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:Describe*",
        "codecommit:Put*",
        "codecommit:Post*",
```

```

        "codecommit:Merge*",
        "codecommit:Test*",
        "codecommit:Update*",
        "codecommit:GitPull",
        "codecommit:GitPush"
    ],
    "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
    ]
}
]
}
```

Si quieres que los usuarios que asuman esta función puedan ver una lista de repositorios en la página de inicio de la CodeCommit consola, añade una declaración adicional a la política, de la siguiente manera:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGet*",
        "codecommit:Create*",
        "codecommit>DeleteBranch",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:Describe*",
        "codecommit:Put*",
        "codecommit:Post*",
        "codecommit:Merge*",
        "codecommit:Test*",
        "codecommit:Update*",
        "codecommit:GitPull",
        "codecommit:GitPush"
      ],
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ]
    }
  ]
}
```



```
    ],
    {
        "Effect": "Allow",
        "Action": "codecommit:ListRepositories",
        "Resource": "*"
    }
]
```

Este acceso facilita a los usuarios que desempeñan esta función con esta política la localización del repositorio al que tienen acceso. Pueden elegir el nombre del repositorio de la lista y se les puede redirigir a la página de inicio del repositorio compartido (Code). Los usuarios no pueden obtener acceso a ninguno de los otros repositorios que ven en la lista, pero pueden ver los repositorios de CuentaA en la página Dashboard (Panel).

Si no quieres permitir que los usuarios que asuman el rol puedan ver una lista de todos los repositorios de AccountA, usa el primer ejemplo de política, pero asegúrate de enviar a esos usuarios un enlace directo a la página principal del repositorio compartido en CodeCommit la consola.

6. Elija Revisar política. El validador de políticas notifica los errores de sintaxis (por ejemplo, si olvida reemplazar el nombre del repositorio y el ID de cuenta de Amazon Web Services de ejemplo por el nombre del repositorio y el ID de su cuenta de Amazon Web Services).
7. En la página de revisión de la política, introduce un nombre para la política (por ejemplo, *CrossAccountAccessForMySharedDemoRepo*). También puede proporcionar una descripción opcional para esta política. Elija Crear política.

Paso 2: Crear una función para el acceso al repositorio en CuentaA

Una vez que haya configurado una política, cree una función que puedan desempeñar los usuarios y grupos de IAM de CuentaB y asocie la política a esa función.

A fin de crear una función para obtener acceso al repositorio

1. En la consola de IAM, seleccione Roles.
2. Elija Crear rol.
3. Seleccione Otra cuenta de Amazon Web Services.

4. En ID de cuenta, introduce el ID de cuenta de Amazon Web Services para AccountB (por ejemplo, **888888888888**). Elija Siguiente: permisos.
5. En Adjuntar políticas de permisos, seleccione la política que creó en el procedimiento anterior (***CrossAccountAccessForMySharedDemoRepo***). Elija Siguiente: Revisar.
6. En Nombre del rol, introduzca un nombre para el rol (por ejemplo, ***MyCrossAccountRepositoryContributorRole***). También puede introducir una descripción opcional para ayudar a otras personas a conocer el objetivo del rol.
7. Elija Crear rol.
8. Abra la función que acaba de crear y copie el ARN de la función (por ejemplo, **`arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole`**). Debe proporcionar este ARN al administrador de CuentaB.

Cross-account acceso al repositorio: acciones para el administrador en AccountB

Para permitir a los usuarios o grupos de Cuenta B obtener acceso a un repositorio en CuentaA, el administrador de Cuenta B debe crear un grupo en Cuenta B. Este grupo debe configurarse con una política que permita a los miembros del grupo desempeñar la función creada por el administrador de CuentaA.

En las siguientes secciones se proporcionan pasos y ejemplos.

Temas

- [Paso 1: Crear un grupo de IAM para el acceso al repositorio de los usuarios de CuentaB](#)
- [Paso 2: Crear una política y añadir usuarios al grupo de IAM](#)

Paso 1: Crear un grupo de IAM para el acceso al repositorio de los usuarios de CuentaB

La forma más sencilla de administrar los usuarios de IAM de Cuenta B que pueden obtener acceso al repositorio de CuentaA consiste en crear un grupo de IAM en Cuenta B que tenga permiso para desempeñar la función en CuentaA y, a continuación, añadir los usuarios de IAM a ese grupo.

A fin de crear un grupo para el acceso al repositorio entre cuentas

1. Inicie sesión en la consola AWS de administración como usuario de IAM con los permisos necesarios para crear grupos y políticas de IAM y administrar los usuarios de IAM en AccountB.
2. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
3. En la consola de IAM, seleccione Grupos.
4. Elija Create New Group (Crear nuevo grupo).
5. En Nombre del grupo, introduzca un nombre para el grupo (por ejemplo,).
DevelopersWithCrossAccountRepositoryAccess Elija Paso siguiente.
6. En Attach Policy, seleccione Next Step. Puede crear la política entre cuentas en el siguiente procedimiento. Termine de crear el grupo.

Paso 2: Crear una política y añadir usuarios al grupo de IAM

Ahora que tiene un grupo, cree la política que permita a los miembros de este grupo desempeñar la función que les proporciona acceso al repositorio en CuentaA. A continuación, añada al grupo los usuarios de IAM de Cuenta B a los que desea permitir el acceso en CuentaA.

A fin de crear una política para el grupo y añadir usuarios al mismo

1. En la consola de IAM, elija Grupos y, a continuación, elija el nombre del grupo que acaba de crear (por ejemplo, *DevelopersWithCrossAccountRepositoryAccess*).
2. Elija la pestaña Permisos. Amplíe Inline Policies (Políticas en línea) y, a continuación, elija el enlace para crear una política en línea [si configura un grupo que ya tiene una política en línea, elija Create Group Policy (Crear política de grupo)].
3. Elija Custom Policy y después Select.
4. En Policy Name (Nombre de política), escriba un nombre para la política (por ejemplo, *AccessPolicyForSharedRepository*).
5. En Policy Document (Documento de políticas), pegue la siguiente política. En **Resource**, sustituya el ARN por el ARN de la política creada por el administrador en AccountA (por ejemplo, arn:aws:iam: ::role/) y, a continuación, seleccione Apply Policy. **111122223333 MyCrossAccountRepositoryContributorRole** Para obtener más información acerca de la política creada por el administrador en CuentaA, consulte [Paso 1: Crear una política de acceso al repositorio en CuentaA](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource":
      "arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole"
  }
}
```

6. Elija la pestaña Users. Seleccione Añadir usuarios al grupo y, a continuación, añada los usuarios de IAM de CuentaB. Por ejemplo, puede añadir al grupo un usuario de IAM con el nombre de usuario. *Saanvi_Sarkar*

Note

Los usuarios de AccountB deben tener acceso mediante programación, que incluye una clave de acceso y una clave secreta, para configurar sus ordenadores locales para acceder al repositorio compartido. CodeCommit Si crea usuarios de IAM, asegúrese de guardar la clave de acceso y la clave secreta. Para garantizar la seguridad de su cuenta de AWS , la clave de acceso secreta solo está accesible cuando la crea.

Cross-account acceso al repositorio: acciones para el usuario del repositorio en AccountB

Para obtener acceso al repositorio en CuentaA, los usuarios del grupo de Cuenta B deben configurar sus equipos locales para obtener acceso al repositorio. En las siguientes secciones se proporcionan pasos y ejemplos.

Temas

- [Paso 1: Configurar el AWS CLI y Git para que un usuario de AccountB acceda al repositorio en AccountA](#)
- [Paso 2: Clona y accede al CodeCommit repositorio en AccountA](#)

Paso 1: Configurar el AWS CLI y Git para que un usuario de AccountB acceda al repositorio en AccountA

No puede utilizar las claves SSH ni las credenciales de Git para obtener acceso a los repositorios de otra cuenta de Amazon Web Services. Los usuarios de AccountB deben configurar sus computadoras para usar git-remote-codecommit (recomendado) o el asistente de credenciales para acceder al repositorio CodeCommit compartido de AccountA. Sin embargo, puede continuar usando las claves SSH o las credenciales de Git al obtener acceso a los repositorios en CuentaB.

Siga estos pasos para configurar el acceso mediante git-remote-codecommit. Si aún no ha instalado git-remote-codecommit, descárguelo desde [git-remote-codecommit](#) en el sitio web de Python Package Index.

Para configurar el AWS CLI y Git para acceso entre cuentas

1. Instálelo AWS CLI en el ordenador local. Consulte las instrucciones del sistema operativo en [Instalación de la AWS CLI](#).
2. Instale Git en el equipo local. Para instalar Git, le recomendamos sitios web como [Git Downloads](#) o [Git for Windows](#).

Note

CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git. Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a su forma de trabajar con CodeCommit ella. Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

3. En el terminal o la línea de comandos, en la ubicación del directorio donde desea clonar el repositorio, ejecute los comandos `git config --local user.name` y `git config --local user.email` para establecer el nombre de usuario y el correo electrónico de las confirmaciones que va a realizar en el repositorio. Por ejemplo:

```
git config --local user.name "Saanvi Sarkar"
git config --local user.email saanvi_sarkar@example.com
```

Estos comandos no devuelven nada, pero el correo electrónico y el nombre de usuario que especifique se asocian a las confirmaciones que realice en el repositorio en CuentaA.

4. Ejecute el comando `aws configure --profile` para configurar un perfil predeterminado que se utiliza para conectar a los recursos en CuentaB. Cuando se le pida, proporcione la clave de acceso y la clave secreta para su usuario de IAM.

 Note

Si ya has instalado el perfil AWS CLI y lo has configurado, puedes saltarte este paso.

Por ejemplo, ejecuta el siguiente comando para crear un AWS CLI perfil predeterminado que utilices para acceder a AWS los recursos de AccountB en el este de EE. UU. (Ohio) (us-east-2):

```
aws configure
```

Cuando se le pida, proporcione la siguiente información:

```
AWS Access Key ID [None]: Your-IAM-User-Access-Key
AWS Secret Access Key ID [None]: Your-IAM-User-Secret-Access-Key
Default region name [None]: us-east-2
Default output format [None]: json
```

5. Vuelva a ejecutar el comando `aws configure --profile` para configurar un perfil con nombre que utilizar al conectarse al repositorio en CuentaA. Cuando se le pida, proporcione la clave de acceso y la clave secreta para su usuario de IAM. Por ejemplo, ejecuta el siguiente comando para crear un nombre de AWS CLI perfil *MyCrossAccountAccessProfile* que utilices para acceder a un repositorio en AccountA en US East (Ohio) (us-east-2):

```
aws configure --profile MyCrossAccountAccessProfile
```

Cuando se le pida, proporcione la siguiente información:

```
AWS Access Key ID [None]: Your-IAM-User-Access-Key
AWS Secret Access Key ID [None]: Your-IAM-User-Secret-Access-Key
Default region name [None]: us-east-2
Default output format [None]: json
```

6. En un editor de texto sin formato, abra el archivo `config`, también conocido como el archivo de configuración de AWS CLI . Según el sistema operativo, este archivo puede estar ubicado `~/.aws/config` en Linux, macOS o Unix, o en `drive:\ Users\ .awsUSERNAME\ config` en Windows.
7. En el archivo, busque la entrada que corresponda al perfil predeterminado configurado para acceder a repositorios de la cuenta B. Debería parecerse a lo que sigue:

```
[default]
region = us-east-2
output = json
```

Agregue `account` a la configuración del perfil. Proporcione el ID de cuenta de AWS de la cuenta B. Por ejemplo:

```
[default]
account = 888888888888
region = us-east-2
output = json
```

8. En el archivo, busque la entrada que corresponda al `MyCrossAccountAccessProfile` perfil que acaba de crear. Debería parecerse a lo que sigue:

```
[profile MyCrossAccountAccessProfile]
region = us-east-2
output = json
```

Agregue `account`, `role_arn` y `source_profile` a la configuración del perfil. Proporcione el ID de la cuenta de Amazon Web Services de la cuenta A, el ARN del rol de la cuenta A que asume para obtener acceso al repositorio en la otra cuenta, así como el nombre de su perfil predeterminado de la AWS CLI en la cuenta B. Por ejemplo:

```
[profile MyCrossAccountAccessProfile]
region = us-east-2
account = 111122223333
role_arn = arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole
source_profile = default
output = json
```

Guarde sus cambios y cierre el editor de texto sin formato.

Paso 2: Clona y accede al CodeCommit repositorio en AccountA

Ejecute `git clone`, `git push` y `git pull` para clonar el repositorio de CodeCommit entre cuentas y utilizarlo para insertar o extraer contenido. También puede iniciar sesión en la consola AWS de administración, cambiar de rol y usar la CodeCommit consola para interactuar con el repositorio de la otra cuenta.

Note

En función de cómo esté configurada la función de IAM, es posible que puedas ver los repositorios en la página predeterminada de CodeCommit. Si no puede ver los repositorios, pida al administrador del repositorio que le envíe por correo electrónico un enlace URL a la página de códigos del repositorio compartido en la consola de CodeCommit. La URL es similar a la siguiente:

```
https://console.aws.amazon.com/codecommit/home?region=us-east-2#/  
repository/MySharedDemoRepo/browse/HEAD/--/
```

Para clonar el repositorio entre cuentas en su equipo local

1. En el terminal o la línea de comandos, en el directorio donde desea clonar el repositorio, ejecute el comando `git clone` con la URL de clonación HTTPS (GRC). Por ejemplo:

```
git clone codecommit://MyCrossAccountAccessProfile@MySharedDemoRepo
```

A menos que especifique lo contrario, el repositorio se clona en un subdirectorio con el mismo nombre que el repositorio.

2. Cambie los directorios al repositorio clonado y añada o realice un cambio en un archivo. Por ejemplo, puedes añadir un archivo con el nombre *NewFile.txt*.
3. Agrega el archivo al registro de cambios del repositorio local, confirma el cambio y envía el archivo al CodeCommit repositorio. Por ejemplo:

```
git add NewFile.txt  
git commit -m "Added a file to test cross-account access to this repository"  
git push
```

Para obtener más información, consulte [Primeros pasos con Git y AWS CodeCommit](#).

Ahora que has añadido un archivo, ve a la CodeCommit consola para ver tu confirmación, revisar los cambios de otros usuarios en el repositorio, participar en las solicitudes de cambios y mucho más.

Para acceder al repositorio multicuenta de la consola CodeCommit

1. Inicia sesión Consola de administración de AWS en AccountB (**888888888888**) como el usuario de IAM al que se le ha concedido acceso multicuenta al repositorio de AccountA.
2. Elija su nombre de usuario en la barra de navegación y, en la lista desplegable, elija Switch Role (Cambiar función).

 Note

Si esta es la primera vez que ha seleccionado esta opción, revise la información de la página y, a continuación, vuelva a elegir Switch Role (Cambiar función).

3. En la página Switch Role (Cambiar función), haga lo siguiente:
 - En Account (Cuenta), introduzca el ID de cuenta para CuentaA (por ejemplo, **111122223333**).
 - En Rol, ingresa el nombre del rol que deseas asumir para acceder al repositorio en AccountA (por ejemplo, **MyCrossAccountRepositoryContributorRole**).
 - En Display Name (Nombre que mostrar), introduzca un nombre fácil de recordar para este rol. Este nombre aparecerá en la consola al desempeñar esta función. También aparecerá en la lista de funciones desempeñadas la próxima vez que desee cambiar funciones en la consola.
 - (Opcional) En Color (Color), elija una etiqueta de color para el nombre de visualización.
 - Elija Switch Role.

Para obtener más información, consulte [Cambiar de función \(Consola de administración de AWS\)](#).

4. Abre la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>

Si la función desempeñada tiene permiso para ver los nombres de los repositorios en CuentaA, verá una lista de repositorios y un mensaje de error informándole de que no tiene permisos para ver su estado. Este es el comportamiento esperado. Elija el nombre del repositorio compartido de la lista.

Si la función desempeñada no tiene permiso para ver los nombres de los repositorios en CuentaA, verá un mensaje de error y una lista en blanco sin repositorios. Pegue el enlace de URL en el repositorio o modifique el enlace de la consola y cambie `/list` por el nombre del repositorio compartido (por ejemplo, `/MySharedDemoRepo`).

5. En Code (Código), busque el nombre del archivo que añadió desde su equipo local. Elija para examinar el código en el archivo y, a continuación, examine el resto del repositorio y comience a usar sus características.

Para obtener más información, consulte [Empezar con AWS CodeCommit](#).

Eliminar un AWS CodeCommit repositorio

Puede utilizar la CodeCommit consola o la AWS CLI para eliminar un CodeCommit repositorio.

Note

Al eliminar un repositorio, no se eliminan las copias locales de ese repositorio (repositorio local). Para eliminar un repositorio local, use el directorio y las herramientas de administración de archivos de su equipo local.

Temas

- [Eliminar un CodeCommit repositorio \(consola\)](#)
- [Eliminar un repositorio local](#)
- [Eliminar un CodeCommit repositorio \(AWS CLI\)](#)

Eliminar un CodeCommit repositorio (consola)

Siga estos pasos para usar la CodeCommit consola para eliminar un CodeCommit repositorio.

Important

Después de eliminar un CodeCommit repositorio, ya no podrá clonarlo en ningún repositorio local o compartido. Tampoco puede extraer ni enviar datos desde ningún repositorio local ni repositorio compartido. Esta acción no se puede deshacer.

1. Abre la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>
2. En Repositories (Repositorios), seleccione el nombre del repositorio que desee eliminar.
3. En el panel de navegación, seleccione Configuración.
4. En la pestaña General en Delete repository (Eliminar repositorio), elija Delete repository (Eliminar repositorio). Introduzca **delete** y luego escriba Delete (Eliminar). El repositorio se eliminará permanentemente.

 Note

Al eliminar el repositorio en, CodeCommit no se elimina ningún repositorio local.

Eliminar un repositorio local

Use el directorio y las herramientas de administración de archivos de su equipo local para eliminar el directorio que contiene el repositorio local.

Al eliminar un repositorio local no se elimina ningún CodeCommit repositorio al que pueda estar conectado.

Eliminar un CodeCommit repositorio (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para utilizar el AWS CLI para eliminar un CodeCommit repositorio, ejecute el delete-repository comando especificando el nombre del CodeCommit repositorio que desee eliminar (con la --repository-name opción).

 Important

Tras eliminar un CodeCommit repositorio, ya no podrá clonarlo en ningún repositorio local o compartido. Tampoco puede extraer ni enviar datos desde ningún repositorio local ni repositorio compartido. Esta acción no se puede deshacer.

 Tip

Para obtener el nombre del CodeCommit repositorio, ejecuta el comando [list-repositories](#).

Por ejemplo, para eliminar un repositorio denominado MyDemoRepo:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el ID del CodeCommit repositorio que se ha eliminado permanentemente aparece en el resultado:

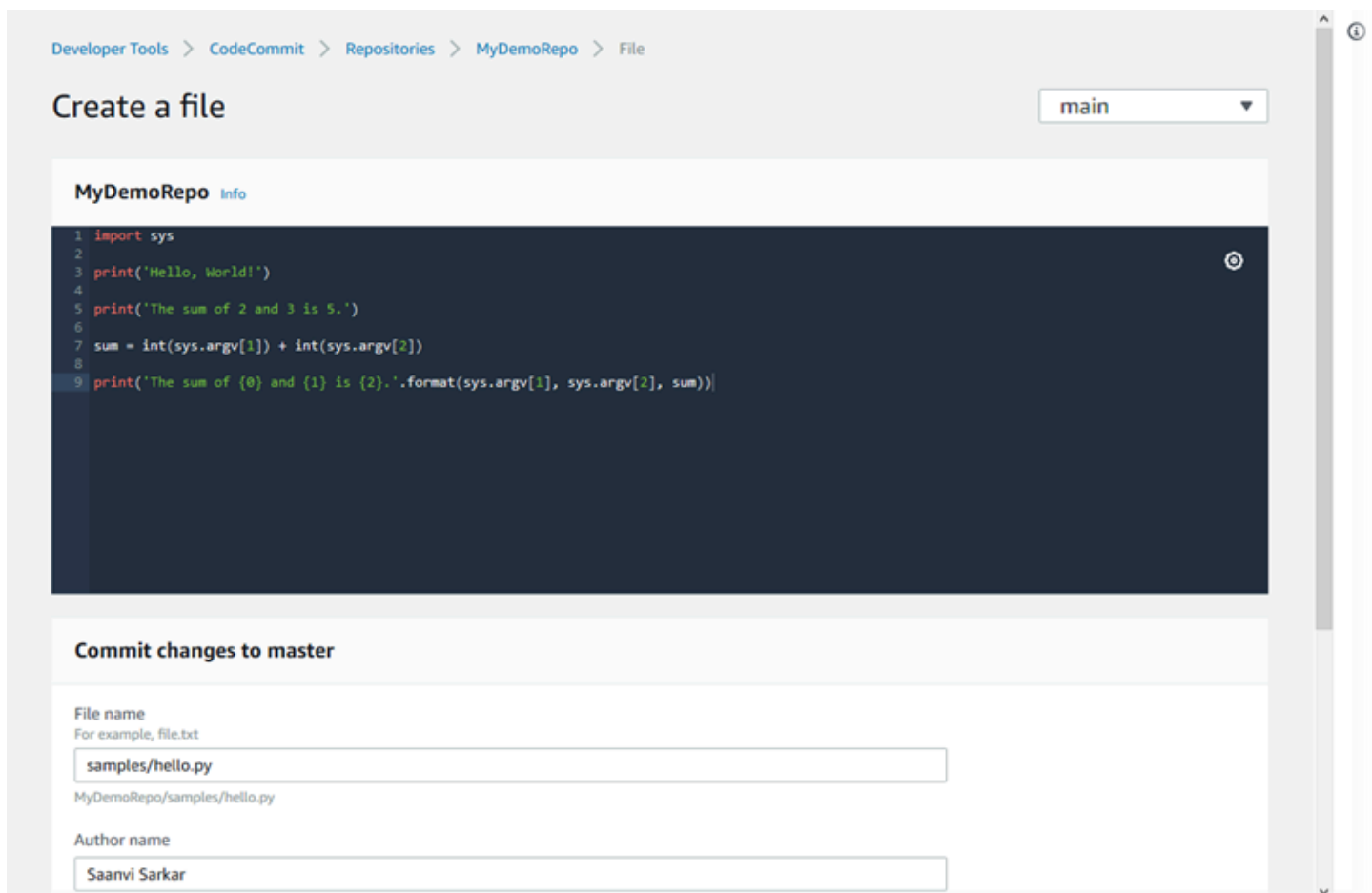
```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

Al eliminar un CodeCommit repositorio, no se elimina ningún repositorio local que pueda estar conectado a él.

Trabajar con archivos en repositorios de AWS CodeCommit

En CodeCommit, un archivo es un dato autónomo con control de versiones disponible para usted y otros usuarios del repositorio y la ramificación donde se almacena el archivo. Puede organizar los archivos de su repositorio con una estructura de directorios, igual que lo haría en un equipo. A diferencia de su equipo, CodeCommit hace automáticamente un seguimiento de todos los cambios realizados en un archivo. Puede comparar y almacenar las diferentes versiones de un archivo en diversas ramificaciones del repositorio.

Para añadir un archivo a un repositorio, puede utilizar un cliente de Git. También puede utilizar la consola CodeCommit, la AWS CLI o la API de CodeCommit.



The screenshot shows the 'Create a file' interface in the AWS CodeCommit console. The breadcrumb navigation at the top reads: Developer Tools > CodeCommit > Repositories > MyDemoRepo > File. A dropdown menu in the top right corner is set to 'main'. The main area displays a code editor for a file named 'MyDemoRepo.py' (with an 'Info' icon). The code is as follows:

```
1 import sys
2
3 print('Hello, World!')
4
5 print('The sum of 2 and 3 is 5.')
6
7 sum = int(sys.argv[1]) + int(sys.argv[2])
8
9 print('The sum of {0} and {1} is {2}.'.format(sys.argv[1], sys.argv[2], sum))
```

Below the code editor, there is a section titled 'Commit changes to master'. It contains two input fields: 'File name' and 'Author name'. The 'File name' field has a hint 'For example, file.txt' and contains the text 'samples/hello.py'. Below this field, the full path 'MyDemoRepo/samples/hello.py' is displayed. The 'Author name' field contains the text 'Saanvi Sarkar'.

Para obtener información acerca de cómo trabajar con otros aspectos del repositorio en CodeCommit, consulte [Trabajar con repositorios](#), [Trabajo con solicitudes de extracción](#), [Uso de las ramificaciones](#), [Uso de confirmaciones](#) y [Trabajo con las preferencias del usuario](#).

Temas

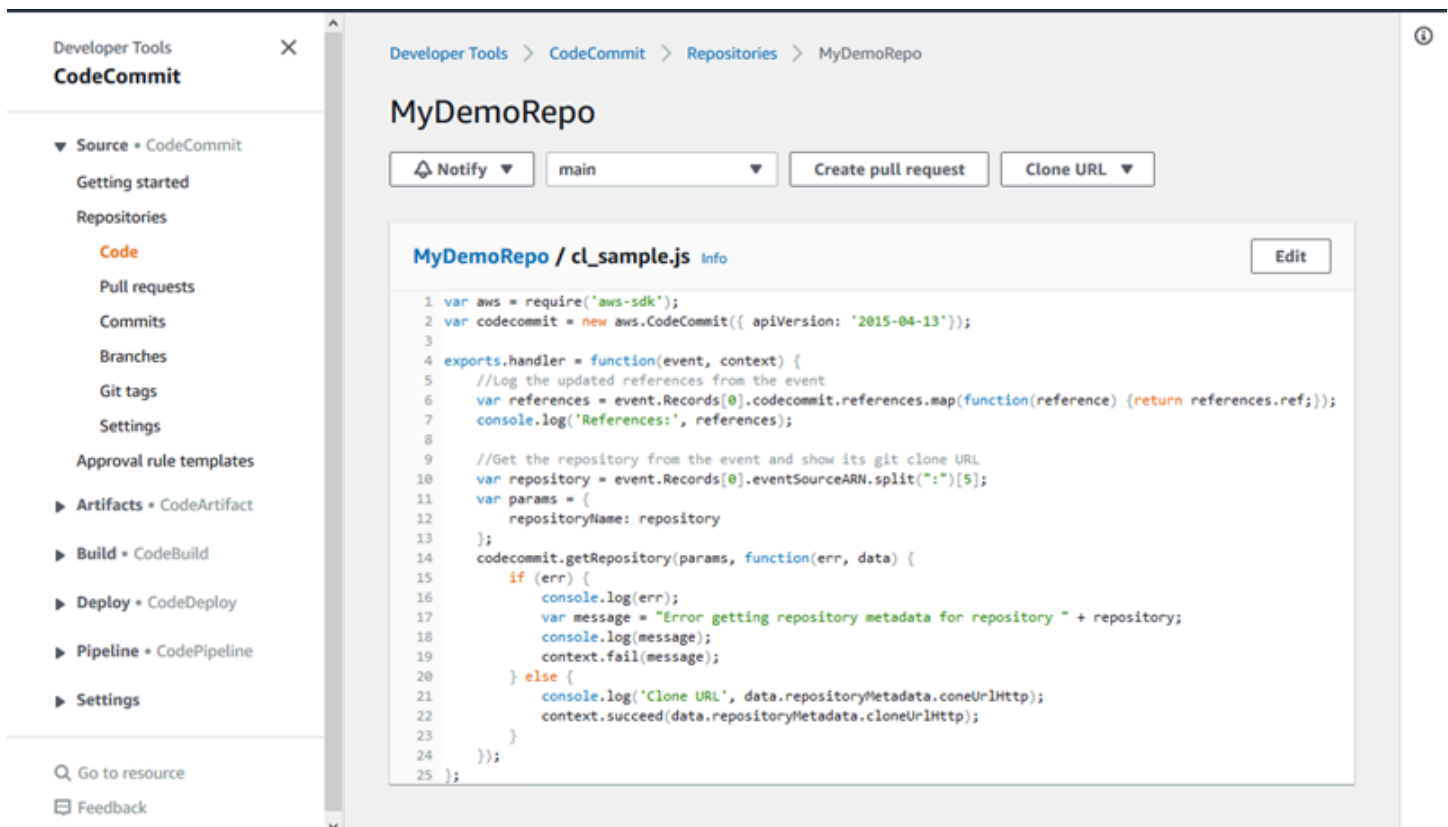
- [Examinar los archivos de un AWS CodeCommit repositorio](#)
- [Crear o añadir un archivo a un repositorio de AWS CodeCommit](#)
- [Editar el contenido de un archivo en un AWS CodeCommit repositorio](#)

Examinar los archivos de un AWS CodeCommit repositorio

Tras conectarse a un CodeCommit repositorio, puede clonarlo en un repositorio local o utilizar la CodeCommit consola para examinar su contenido. En este tema se describe cómo usar la CodeCommit consola para examinar el contenido de un CodeCommit repositorio.

Note

Para CodeCommit los usuarios activos, navegar por el código desde la CodeCommit consola es gratuito. Para obtener información acerca de cuándo podrían aplicarse los cargos, consulte la sección [Pricing](#).



Navegue por un CodeCommit repositorio

Puede utilizar la CodeCommit consola para revisar los archivos contenidos en un repositorio o para leer rápidamente el contenido de un archivo.

Para examinar el contenido de un repositorio

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En la página Repositories (Repositorios), en la lista de repositorios, seleccione el repositorio que desea examinar.
3. En la vista Code (Código), revise el contenido de la ramificación predeterminada del repositorio.

Para cambiar a la vista de otra ramificación o etiqueta, elija el botón Selector de vista.

Seleccione el nombre de una ramificación o etiqueta en la lista desplegable o, en el cuadro de filtro, introduzca el nombre de la ramificación o etiqueta. A continuación, selecciónela.

4. Realice una de las siguientes acciones:
 - Para ver el contenido de un directorio, elíjalo en la lista. Puede elegir cualquiera de los directorios de la lista de navegación para volver a esa vista de directorio. También puede utilizar la flecha arriba que aparece en la parte superior de la lista de directorios.
 - Para ver el contenido de un archivo, elíjalo en la lista. Si el archivo es mayor que el límite de objetos de confirmación, no se mostrará en la consola y, en su lugar, deberá verlo en un repositorio local. Para obtener más información, consulte [Cuotas](#). Para salir de la vista de archivos, en la barra de navegación por códigos, seleccione el directorio que desea ver.

Note

No todos los archivos binarios se pueden ver en la consola. Si selecciona un archivo binario y puede visualizarse, aparecerá un mensaje de advertencia que le pedirá que confirme que desea ver el contenido. Para ver el archivo, elija Show file contents. Si no desea ver el archivo, en la barra de navegación por códigos, seleccione el directorio que desea ver.

Si elige un archivo markdown (.md), utilice los botones Rendered Markdown y Markdown Source para cambiar entre las vistas representadas y de sintaxis. Para obtener más información, consulte [Usar marcado en la consola](#).

Crear o añadir un archivo a un repositorio de AWS CodeCommit

Puede usar la consola de CodeCommit, la AWS CLI o un cliente de Git para añadir un archivo a un repositorio. Puede cargar un archivo desde su equipo local en el repositorio, o puede usar el editor de código en la consola para crear el archivo. El editor es una forma rápida y fácil de añadir un archivo sencillo, como un archivo `readme.md`, a una ramificación en un repositorio.

Upload a file

MyDemoRepo [Info](#)

Name	Size	Actions
Upload file Choose a file to upload.  Choose file		

Commit changes to master

Author name

Email address

Commit message - optional
A default commit message will be used if you do not provide one.

Cancel Commit changes

Temas

- [Crear o cargar un archivo \(Consola\)](#)
- [Añadir un archivo \(AWS CLI\)](#)
- [Añadir de un archivo \(Git\)](#)

Crear o cargar un archivo (Consola)

Puede usar la consola de CodeCommit para crear un archivo y añadirlo a una ramificación en un repositorio de CodeCommit. Como parte de la creación del archivo, puede proporcionar su nombre de usuario y una dirección de correo electrónico. También puede añadir un mensaje de confirmación para que otros usuarios sepan quién añadió el archivo y por qué. También puede cargar un archivo directamente desde su equipo local en una ramificación en un repositorio.

Para añadir un archivo a un repositorio

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio donde desea añadir un archivo.
3. En la vista Code (Código), elija la ramificación en la que desea añadir el archivo. De forma predeterminada, el contenido de la ramificación predeterminada se muestra al abrir la vista Code (Código).

Para cambiar a la vista de otra ramificación, elija el botón Selector de vista. Seleccione el nombre de una ramificación en la lista desplegable o, en el cuadro de filtro, introduzca el nombre de la ramificación. A continuación, selecciónela.

4. Elija Add file (Añadir archivo) y, a continuación, elija una de las siguientes opciones:
 - Para usar el editor de código a fin de crear el contenido de un archivo y añadirlo al repositorio, elija Create file (Crear archivo).
 - Para cargar un archivo desde el equipo local en el repositorio, elija Upload file (Cargar archivo).
5. Proporcione información a otros usuarios acerca de quién añadió este archivo al repositorio y por qué.
 - En Author name (Nombre del autor), escriba su nombre. Este nombre se usa como nombre del autor y nombre del confirmador en la información de confirmación. CodeCommit utiliza de forma predeterminada su nombre de usuario de IAM o una derivación del inicio de sesión de su consola como nombre del autor.
 - En Dirección de correo electrónico, introduzca una dirección de correo electrónico para que otros usuarios del repositorio puedan contactar con usted en relación con este cambio.
 - En Commit message (Mensaje de confirmación), introduzca una breve descripción. Este paso es opcional, pero muy recomendable. De lo contrario, se utiliza un mensaje de confirmación predeterminado.
6. Realice una de las siguientes acciones:
 - Si carga un archivo, elija este en su equipo local.
 - Si crea un archivo, introduzca el contenido que desea añadir en el editor de código y proporcione un nombre para el archivo.
7. Seleccione Confirmar cambios.

Añadir un archivo (AWS CLI)

Puede usar la AWS CLI y el comando `put-file` para añadir un archivo en un repositorio de CodeCommit. También puede usar el comando `put-file` para añadir una estructura de directorios o rutas para el archivo.

Note

Para utilizar los comandos de AWS CLI en CodeCommit, instale la AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para añadir un archivo a un repositorio

1. En su equipo local, cree el archivo que desea añadir al repositorio de CodeCommit.
2. En el terminal o en la línea de comandos, ejecute el comando `put-file` especificando lo siguiente:
 - El repositorio en el que desea añadir el archivo.
 - La ramificación en la que desea añadir el archivo.
 - El ID de confirmación completo de la confirmación más reciente realizada a esa ramificación, también conocido como confirmación TIP o HEAD.
 - La ubicación local del archivo. La sintaxis usada para esta ubicación varía dependiendo de su sistema operativo local.
 - El nombre del archivo que desea añadir, incluida la ruta en la que se almacena el archivo actualizado en el repositorio, si la hay.
 - El nombre de usuario y el correo electrónico que desea asociar a este archivo.
 - Un mensaje de confirmación en el que se explica por qué añadió este archivo.

El nombre de usuario, la dirección de correo electrónico y el mensaje de confirmación son opcionales, pero ayudan a otros usuarios a saber quién realizó el cambio y por qué. Si no proporciona un nombre de usuario, CodeCommit utiliza de forma predeterminada el nombre de usuario de IAM o una variante del inicio de sesión de la consola como nombre del autor.

Por ejemplo, para añadir un archivo denominado *ExampleSolution.py* a un repositorio llamado *MyDemoRepo* en una ramificación denominada *feature-randomizationfeature* cuya confirmación más reciente tiene un ID de *4c925148EXAMPLE*:

```
aws codecommit put-file --repository-name MyDemoRepo --branch-name feature-randomizationfeature --file-content file://MyDirectory/ExampleSolution.py --file-path /solutions/ExampleSolution.py --parent-commit-id 4c925148EXAMPLE --name "María García" --email "maría_garcía@example.com" --commit-message "I added a third randomization routine."
```

Note

Al añadir archivos binarios, asegúrese de usar `fileb://` para especificar la ubicación local del archivo.

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

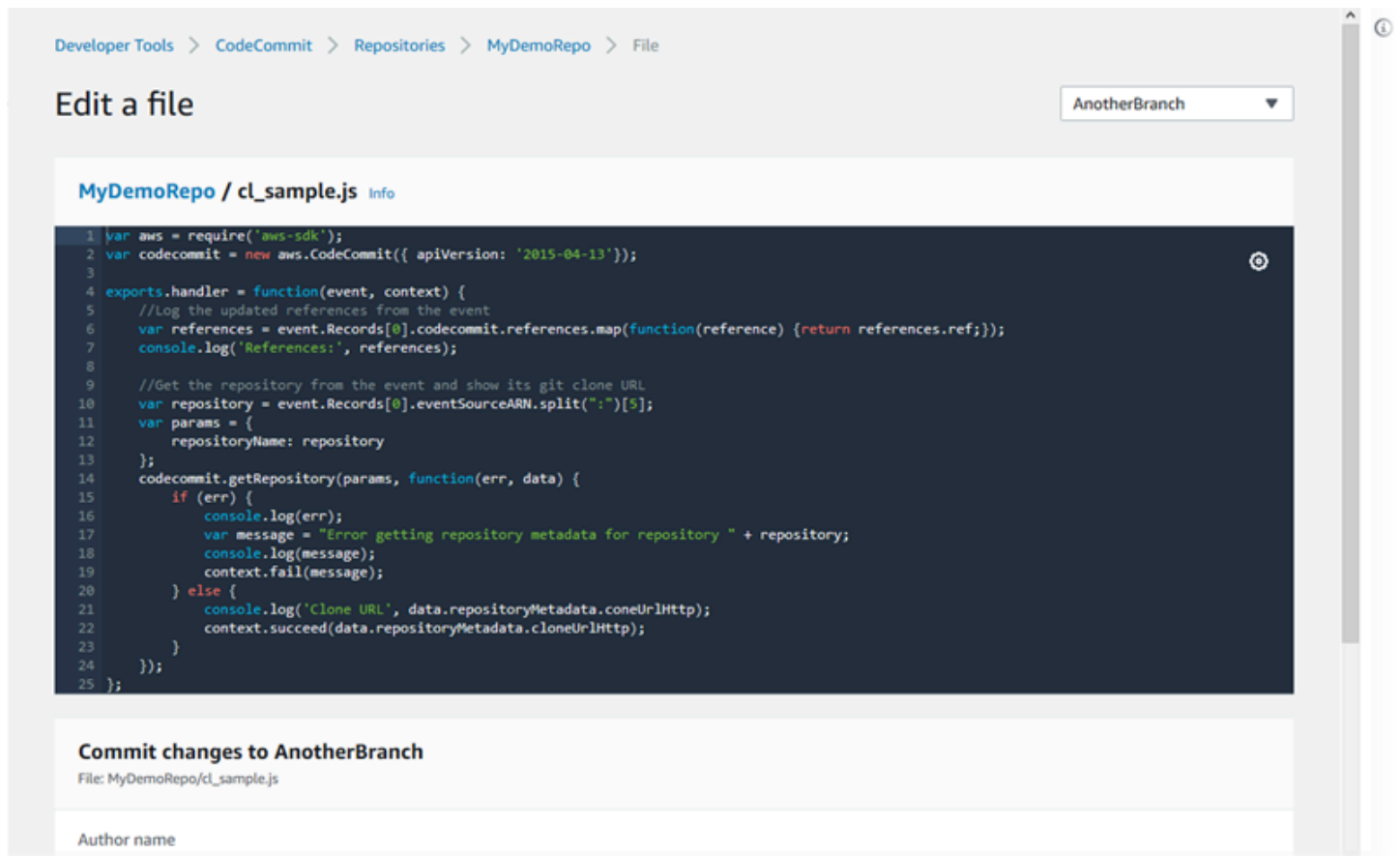
```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Añadir de un archivo (Git)

Puede añadir archivos en un repositorio local y enviar los cambios a un repositorio de CodeCommit. Para obtener más información, consulte [Primeros pasos con Git y AWS CodeCommit](#).

Editar el contenido de un archivo en un AWS CodeCommit repositorio

Puedes usar la CodeCommit consola o un cliente Git para editar el contenido de un archivo en un CodeCommit repositorio. AWS CLI



Temas

- [Editar un archivo \(consola\)](#)
- [Editar o eliminar un archivo \(AWS CLI\)](#)
- [Editar un archivo \(Git\)](#)

Editar un archivo (consola)

Puede usar la CodeCommit consola para editar un archivo que se ha agregado a una rama de un CodeCommit repositorio. Como parte de la edición del archivo, puede proporcionar su nombre de usuario y una dirección de correo electrónico. También puede añadir un mensaje de confirmación para que otros usuarios sepan quién realizó el cambio y por qué.

Para editar un archivo en un repositorio

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio donde desea editar un archivo.

3. En la vista Code (Código), elija la ramificación en la que desea editar el archivo. De forma predeterminada, el contenido de la ramificación predeterminada se muestra al abrir la vista Code (Código).

Para cambiar a la vista de otra ramificación, elija el botón Selector de vista. Seleccione el nombre de una ramificación en la lista desplegable o, en el cuadro de filtro, introduzca el nombre de la ramificación. A continuación, selecciónela.

4. Desplácese por el contenido de la ramificación y elija el archivo que desea editar. En la vista de archivo, elija Edit (Editar).

Note

Si elige un archivo binario, aparecerá un mensaje de advertencia que le pedirá que confirme que desea ver el contenido. No debe utilizar la CodeCommit consola para editar archivos binarios.

5. Edite el archivo y proporcione información a otros usuarios acerca de quién realizó este cambio y por qué.
 - En Author name (Nombre del autor), escriba su nombre. Este nombre se utiliza como nombre del autor y como nombre del responsable en la información de confirmación. CodeCommit de forma predeterminada, utiliza su nombre de usuario de IAM o una derivación del inicio de sesión de la consola como nombre de autor.
 - En Dirección de correo electrónico, introduzca una dirección de correo electrónico para que otros usuarios del repositorio puedan contactar con usted en relación con este cambio.
 - En Commit message (Mensaje de confirmación), introduzca una breve descripción de sus cambios.
6. Elija Commit changes (Confirmar cambios) para guardar sus cambios en el archivo y confirmar los cambios en el repositorio.

Editar o eliminar un archivo (AWS CLI)

Puede utilizar el `put-file` comando AWS CLI and para realizar cambios en un archivo de un repositorio. CodeCommit También puede usar el comando `put-file` para añadir una estructura de directorios o rutas para el archivo cambiado si desea almacenar este en una ubicación distinta de la original. Si desea eliminar un archivo totalmente, puede usar el comando `delete-file`.

 Note

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para editar un archivo en un repositorio

1. Mediante una copia local del archivo, realice los cambios que desee añadir al repositorio de CodeCommit.
2. En el terminal o en la línea de comandos, ejecute el comando `put-file` especificando lo siguiente:
 - El repositorio en el que desea añadir el archivo editado.
 - La ramificación en la que desea añadir el archivo editado.
 - El ID de confirmación completo de la confirmación más reciente realizada a esa ramificación, también conocido como confirmación TIP o HEAD.
 - La ubicación local del archivo.
 - El nombre del archivo actualizado que desea añadir, incluida la ruta en la que se almacena el archivo actualizado en el repositorio, si la hay.
 - El nombre de usuario y el correo electrónico que desea asociar a este cambio en el archivo.
 - Un mensaje de confirmación que explica el cambio que realizó.

El nombre de usuario, la dirección de correo electrónico y el mensaje de confirmación son opcionales, pero ayudan a otros usuarios a saber quién realizó el cambio y por qué. Si no proporciona un nombre de usuario, se utilizará de CodeCommit forma predeterminada su nombre de usuario de IAM o una derivación del inicio de sesión de la consola.

Por ejemplo, para añadir las modificaciones realizadas a un archivo con el nombre de un repositorio *ExampleSolution.py* con el nombre *MyDemoRepo* de una rama *feature-randomizationfeature* cuya confirmación más reciente tenga un identificador de: *4c925148EXAMPLE*

```
aws codecommit put-file --repository-name MyDemoRepo --branch-name feature-  
randomizationfeature --file-content file://MyDirectory/ExampleSolution.py --file-  
path /solutions/ExampleSolution.py --parent-commit-id 4c925148EXAMPLE --name "María"
```

```
García" --email "maría_garcía@example.com" --commit-message "I fixed the bug Mary found."
```

Note

Si desea añadir un archivo binario modificado, asegúrese de usar `--file-content` con la notación **fileb://MyDirectory/MyFile.raw**.

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Para eliminar un archivo, utilice el comando `delete-file`. Por ejemplo, para eliminar un archivo nombrado *README.md* en una rama denominada *main* con el ID de confirmación más reciente *c5709475EXAMPLE* de un repositorio llamado *MyDemoRepo*:

```
aws codecommit delete-file --repository-name MyDemoRepo --branch-name main --file-path README.md --parent-commit-id c5709475EXAMPLE
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "blobId": "559b44fEXAMPLE",
  "commitId": "353cf655EXAMPLE",
  "filePath": "README.md",
  "treeId": "6bc824cEXAMPLE"
}
```

Editar un archivo (Git)

Puedes editar los archivos de un repositorio local y enviar los cambios a un CodeCommit repositorio. Para obtener más información, consulte [Primeros pasos con Git y AWS CodeCommit](#).

Trabajar con solicitudes de cambios en AWS CodeCommit repositorios

Una solicitud de extracción es la forma principal de que usted y otros usuarios de un repositorio puedan revisar, realizar comentarios y combinar los cambios en el código de una ramificación a otra. Puede utilizar las solicitudes de extracción para revisar de forma colaborativa los cambios en el código relacionados con pequeños cambios o correcciones, incorporación de características importantes o versiones nuevas del software publicado. A continuación se describe un posible flujo de trabajo para una solicitud de extracción:

Li Juan, un desarrollador que trabaja en un repositorio denominado MyDemoRepo, quiere trabajar en una nueva función para una próxima versión de un producto. Para mantener su trabajo separado del código listo para producción, crea una rama a partir de la rama predeterminada y le da un nombre. *feature-randomizationfeature* Escribe el código, realiza confirmaciones y envía el código de las nuevas características a esta ramificación. Quiere que otros usuarios del repositorio revisen la calidad del código antes de combinar sus cambios en la ramificación predeterminada. Para ello, crea una solicitud de extracción. La solicitud de extracción contiene la comparación entre la ramificación de trabajo y la ramificación del código donde desea combinar sus cambios (en este caso, la ramificación predeterminada). También puede crear una regla de aprobación que requiera un número específico de usuarios para aprobar su solicitud de extracción. Incluso puede especificar un grupo de usuarios de aprobación. Otros usuarios revisan el código y los cambios, y añaden comentarios y sugerencias. Podría actualizar la ramificación de trabajo varias veces con cambios en el código en respuesta a los comentarios realizados. Sus cambios se incorporan a la solicitud de cambios cada vez que los envía a esa rama. CodeCommit También podría incorporar los cambios realizados en la ramificación de destino prevista mientras esté abierta la solicitud de extracción para que los usuarios puedan estar seguros de que están revisando todos los cambios propuestos en el contexto adecuado. Cuando ella y sus revisores están satisfechos, y las condiciones para las reglas de aprobación (si las hay) se han cumplido, ella o uno de sus revisores combinan su código y cierra la solicitud de extracción.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > Create pull request

Create pull request

Destination: main Source: bugfix-1236 Compare Cancel

Mergeable
There are currently no conflicts between bugfix-1236 and main. You can close this pull request by merging it in the AWS CodeCommit console.

Details Create pull request

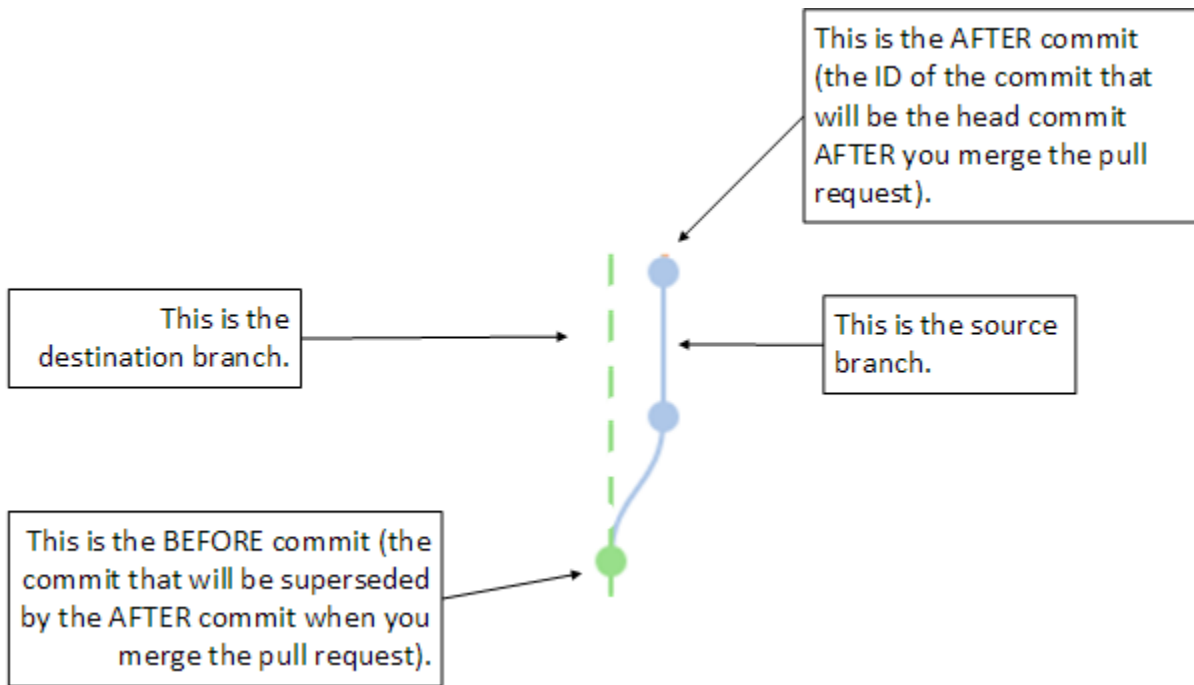
Title
Review changes for bugfix-1236
150 characters maximum

Description - optional Preview markdown Learn more

I've added some code for the bucket creation issue. Please review by Tuesday.

Changes Commits

Las solicitudes de extracción requieren dos ramificaciones: una de origen, que contiene el código que se desea revisar, y otra de destino, donde se combina 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. La elección de la estrategia de fusión afecta a los detalles de cómo se combinan las confirmaciones entre las ramas de origen y destino en la CodeCommit consola. Para obtener más información sobre las estrategias de fusión en CodeCommit, consulte [Combinación de una solicitud de extracción \(consola\)](#).



La solicitud de extracción muestra las diferencias que existen entre el extremo de la ramificación de origen y la última confirmación de la ramificación de destino cuando se va a crear la solicitud de extracción, lo que permite a los usuarios ver y realizar comentarios sobre los cambios. Puede actualizar la solicitud de extracción en respuesta a los comentarios realizados, mediante la confirmación y el envío de estos cambios a la ramificación de origen.

Developer Tools
CodeCommit

Source • CodeCommit

- Getting started
- Repositories
- Code
- Pull requests**
- Commits
- Branches
- Tags
- Settings
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline

Mergeable Learn more

Details Activity **Changes** Commits

< Page 1 of 1 > Go to file

Hide whitespace changes Unified Split

ahs_count.py

Browse file contents Comment on file

```

*** @ -5,6 +5,6 @@
5
6 total = (ess + z)
7 ahs = "Number of alveolar hissing sibilants: {}"
8 - print(ahs.format(total))
8 + print(alv.format(total))

```

New comment Preview markdown Learn more

You've switched back to the old variable, which won't work. This should be ahs.

Save Cancel

Cuando se haya revisado el código y se hayan cumplido los requisitos de la regla de aprobación (si los hay), puede cerrar la solicitud de extracción de una de las siguientes maneras:

- Combinando las ramificaciones localmente y enviar los cambios. Esto cierra la solicitud automáticamente si se utiliza la estrategia de fusión rápida y no hay conflictos de fusión.
- Usa la AWS CodeCommit consola para cerrar la solicitud de extracción sin fusionar, resolver los conflictos en una fusión o, si no hay conflictos, cerrar y fusionar las ramas mediante una de las estrategias de fusión disponibles.
- Utilice la AWS CLI.

Antes de crear una solicitud de extracción:

- Asegúrese de que ha confirmado y enviado los cambios de código que desea revisar para una ramificación (la ramificación de origen).
- Configure las notificaciones del repositorio para que otros usuarios puedan recibir notificaciones sobre la solicitud de extracción y los cambios que se realicen en ella. (Este paso es opcional, pero recomendable).
- Cree y asocie plantillas de reglas de aprobación con el repositorio, de modo que las reglas de aprobación se creen automáticamente para las solicitudes de extracción con el fin de ayudar a garantizar la calidad del código. Para obtener más información, consulte [Trabajar con plantillas de reglas de aprobación](#).

Las solicitudes de extracción son más efectivas cuando se han configurado usuarios de IAM para los usuarios del repositorio en la cuenta de Amazon Web Services. Es más fácil identificar qué usuario ha hecho cada comentario. La otra ventaja es que los usuarios de IAM pueden usar las credenciales de Git para acceder al repositorio. Para obtener más información, consulte [Paso 1: Configuración inicial para CodeCommit](#). Puede utilizar las solicitudes de extracción con otros tipos de usuario, incluidos los usuarios con acceso federado.

Para obtener información sobre cómo trabajar con otros aspectos del repositorio en CodeCommit [Trabajar con repositorios](#), consulte [Trabajar con plantillas de reglas de aprobación](#), [Trabajar con archivos](#), [Uso de confirmaciones](#) [Uso de las ramificaciones](#), y [Trabajo con las preferencias del usuario](#).

Temas

- [Crear una solicitud de extracción](#)
- [Crear una regla de aprobación para una solicitud de extracción](#)
- [Ver solicitudes de cambios en un AWS CodeCommit repositorio](#)

- [Revisión de una solicitud de extracción](#)
- [Actualización de una solicitud de extracción](#)
- [Editar o eliminar una regla de aprobación para una solicitud de extracción](#)
- [Anular reglas de aprobación en una solicitud de extracción](#)
- [Combinación de una solicitud de extracción en un repositorio de AWS CodeCommit](#)
- [Resolver conflictos en una solicitud de extracción en un repositorio de AWS CodeCommit](#)
- [Cerrar una solicitud de extracción en un AWS CodeCommit repositorio](#)

Crear una solicitud de extracción

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. Después de confirmar y enviar esos cambios al repositorio, puede crear una solicitud de extracción que compare el contenido de dicha ramificación (la ramificación de origen) con la ramificación en la que desea combinar los cambios después de que la solicitud de extracción se cierre (la ramificación de destino).

Puede utilizar la consola AWS CodeCommit o la AWS CLI para crear las solicitudes de extracción del repositorio.

Temas

- [Crear una solicitud de extracción \(Consola\)](#)
- [Crear una solicitud de extracción \(AWS CLI\)](#)

Crear una solicitud de extracción (Consola)

Puede utilizar la consola de CodeCommit para crear una solicitud de extracción en un repositorio de CodeCommit. Si el repositorio está [configurado con notificaciones](#), los usuarios que estén suscritos recibirán un mensaje de correo electrónico cuando se cree una solicitud de extracción.

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio donde desea crear una solicitud de extracción.
3. En el panel de navegación, elija Pull Requests.

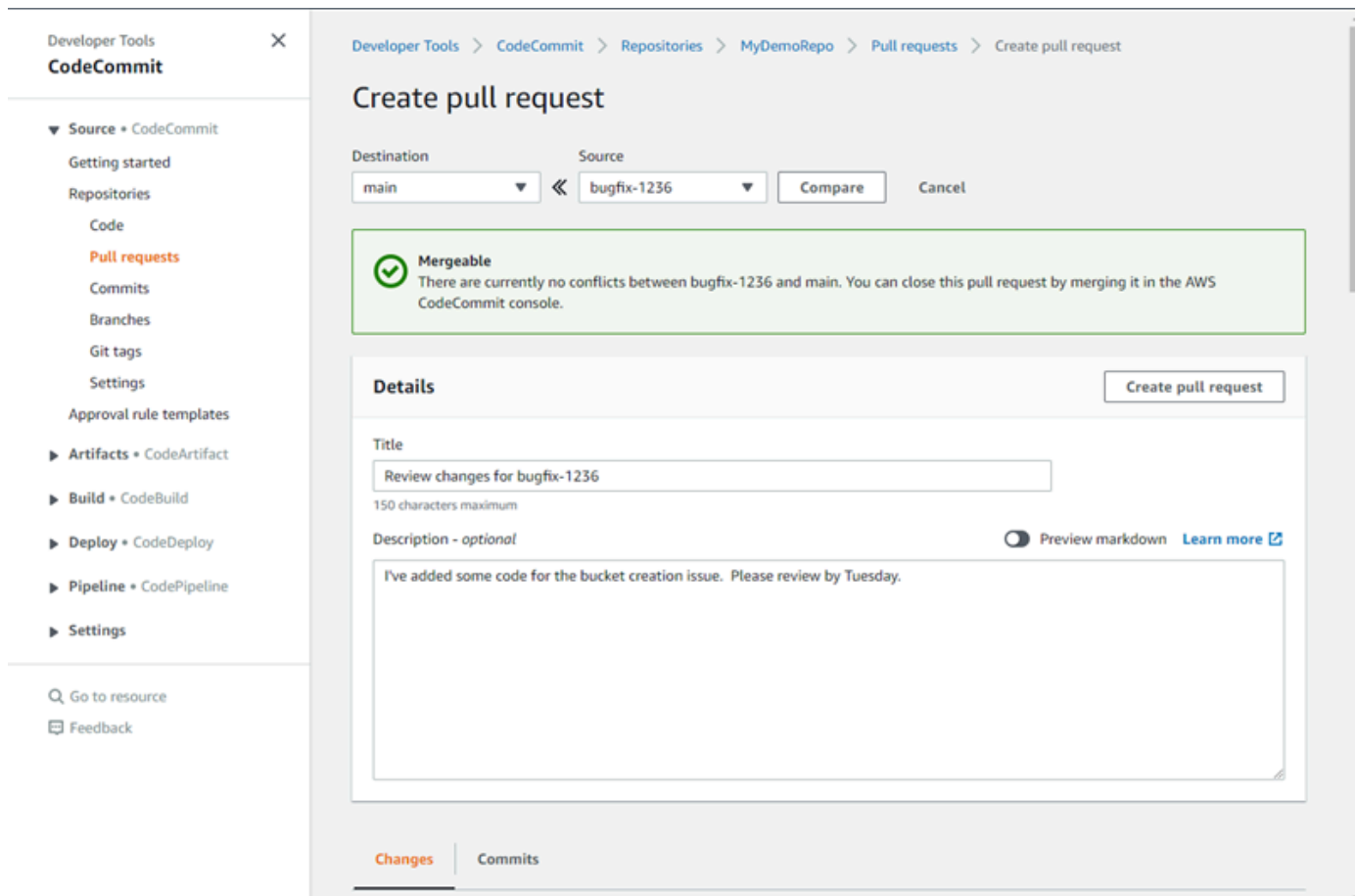
Tip

También puede crear solicitudes de extracción en Branches (Ramificaciones) y Code (Código).

4. Elija Create pull request.

Pull request	Author	Destination	Last activity	Status	Approval status
31: testing this	Saanvi_Sarkar	preprod	4 minutes ago	Open	No approval rules
25: Updated some of our Java samples	Li_Juan	main	5 minutes ago	Open	0 of 1 rules satisfied
29: Changing duplicate value	Li_Juan	main	19 minutes ago	Open	0 of 1 rules satisfied
22: Test pull request	Saanvi_Sarkar	preprod	20 minutes ago	Open	No approval rules
28: Changes to some of our code samples	Li_Juan	main	1 month ago	Open	0 of 1 rules satisfied
20: A bugfix to add missing examples for S3	Saanvi_Sarkar	main	1 month ago	Open	0 of 1 rules satisfied

- En Create pull request, en Source, seleccione la ramificación que contiene los cambios que desea revisar.
- En Destination (Destino), seleccione la ramificación en la que desea combinar los cambios de código cuando se cierre la solicitud de extracción.
- Elija Compare. Se realiza una comparación entre las dos ramificaciones y se muestran las diferencias entre ellas. También se realiza un análisis para determinar si las dos ramificaciones se pueden fusionar automáticamente cuando se cierre la solicitud de extracción.
- Revise la información de la comparación y los cambios para asegurarse de que la solicitud de extracción contiene los cambios y las confirmaciones que desea revisar. En caso contrario, ajuste sus opciones para las ramificaciones de origen y destino y elija Compare de nuevo.
- Cuando esté satisfecho con los resultados de la comparación de la solicitud de extracción, introduzca un título breve pero descriptivo para esta revisión en el campo Title (Título). Este es el título que aparece en la lista de solicitudes de extracción para el repositorio.
- (Opcional) En Description (Descripción), introduzca información sobre esta revisión y cualquier otro dato que pueda ser de utilidad a los revisores.
- Seleccione Crear.



Su solicitud de extracción aparece en la lista de las solicitudes de extracción del repositorio. Si ha [configurado notificaciones](#), los suscriptores al tema de Amazon SNS recibirán un mensaje de correo electrónico para informarles de la solicitud de extracción recién creada.

Crear una solicitud de extracción (AWS CLI)

Para utilizar los comandos de AWS CLI en CodeCommit, instale la AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar la AWS CLI para crear una solicitud de extracción en un repositorio de CodeCommit

- Ejecute el comando `create-pull-request`, especificando:
 - El nombre de la solicitud de extracción (con la opción `--title`).
 - La descripción de la solicitud de extracción (con la opción `--description`).
 - Una lista de destinos para el comando `create-pull-request`, entre las que se incluyen:

- El nombre del repositorio de CodeCommit en que se va a crear la solicitud de extracción (con el atributo `repositoryName`).
- El nombre de la ramificación que contiene los cambios de código que desea revisar, también conocida como ramificación de origen (con el atributo `sourceReference`).
- (Opcional) El nombre de la ramificación en la que desea fusionar los cambios de código, también conocida como ramificación de destino, si no desea fusionarlos en la ramificación predeterminada (con el atributo `destinationReference`).
- Un token de idempotencia único generado por el cliente (con la opción `--client-request-token`).

En este ejemplo se crea una solicitud de extracción denominada *Pronunciation difficulty analyzer* (Analizador de dificultades de pronunciación) con la una descripción *Please review these changes by Tuesday* (Revisar estos cambios antes del martes), dirigida a la ramificación de origen *jane-branch*. La solicitud de extracción se va a combinar en la ramificación predeterminada *principal* en un repositorio de CodeCommit denominado *MyDemoRepo*:

```
aws codecommit create-pull-request --title "Pronunciation difficulty analyzer"
--description "Please review these changes by Tuesday" --client-request-token
123Example --targets repositoryName=MyDemoRepo,sourceReference=jane-branch
```

Crear una regla de aprobación para una solicitud de extracción

La creación de reglas de aprobación para las solicitudes de extracción ayuda a garantizar la calidad del código, porque exige a los usuarios que aprueben la solicitud de extracción antes de que el código se pueda combinar en la ramificación de destino. Puede especificar el número de usuarios que deben aprobar una solicitud de extracción. También puede especificar un grupo de usuarios de aprobación para la regla. Si lo hace, solo las aprobaciones de esos usuarios se contabilizan en el número de aprobaciones necesarias para la regla.

Note

Además, puede crear plantillas de reglas de aprobación, que pueden ayudarle a automatizar la creación de reglas de aprobación en todos los repositorios que se apliquen a cada solicitud de extracción. Para obtener más información, consulte [Trabajar con plantillas de reglas de aprobación](#).

Puedes usar la AWS CodeCommit consola o la AWS CLI para crear reglas de aprobación para tu repositorio.

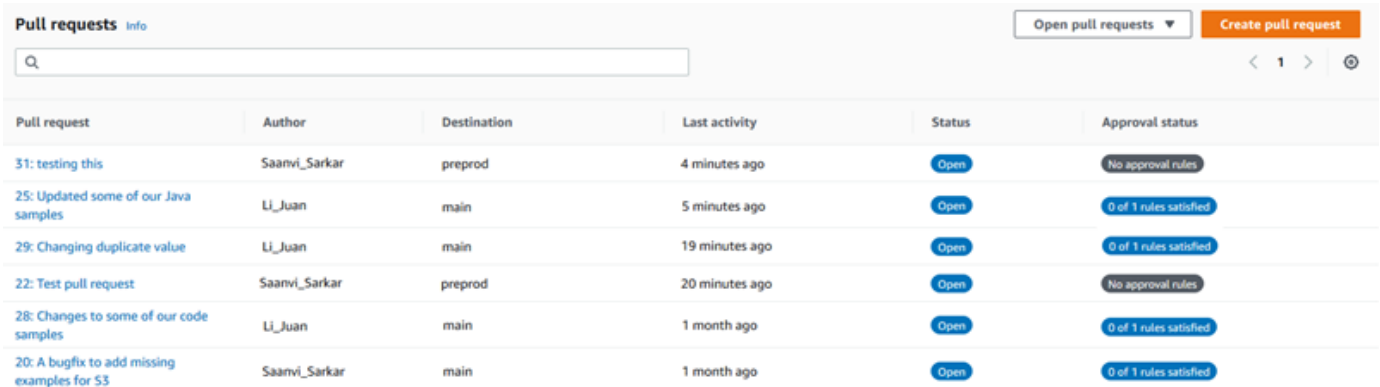
Temas

- [Crear una regla de aprobación para una solicitud de extracción \(consola\)](#)
- [Crear una regla de aprobación para una solicitud de extracción \(AWS CLI\)](#)

Crear una regla de aprobación para una solicitud de extracción (consola)

Puedes usar la CodeCommit consola para crear una regla de aprobación para una solicitud de extracción en un CodeCommit repositorio.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio donde desea crear una regla de aprobación para una solicitud de extracción.
3. En el panel de navegación, elija Pull Requests.
4. Seleccione en la lista la solicitud de extracción cuya regla de aprobación desea crear. Solo puede crear reglas de aprobación para solicitudes de extracción abiertas.



Pull request	Author	Destination	Last activity	Status	Approval status
31: testing this	Saanvi_Sarkar	preprod	4 minutes ago	Open	No approval rules
25: Updated some of our Java samples	Li_Juan	main	5 minutes ago	Open	0 of 1 rules satisfied
29: Changing duplicate value	Li_Juan	main	19 minutes ago	Open	0 of 1 rules satisfied
22: Test pull request	Saanvi_Sarkar	preprod	20 minutes ago	Open	No approval rules
28: Changes to some of our code samples	Li_Juan	main	1 month ago	Open	0 of 1 rules satisfied
20: A bugfix to add missing examples for S3	Saanvi_Sarkar	main	1 month ago	Open	0 of 1 rules satisfied

5. En la solicitud de extracción, elija Approvals (Aprobaciones) y elija Create approval rule (Crear regla de aprobación).
6. En Rule name (Nombre de regla), asigne a la regla un nombre descriptivo que permita saber para qué sirve. Por ejemplo, si desea requerir que dos personas aprueben una solicitud de extracción para poder combinarla, puede asignar a la regla el nombre **Require two approvals before merge**.

Note

El nombre de una regla de aprobación no se puede cambiar una vez se ha creado.

En Number of approvals needed (Número de aprobaciones necesarias), introduzca el número que desee. El valor predeterminado es 1.

Create approval rule

Rule details

Rule name
Require two approvals before merge

Number of approvals needed
2

Approval pool members - *optional*
If approval pool members are specified, only approvals from these members will count toward satisfying this rule. Use a wildcard to match multiple approvers with one value.

Add

Cancel Submit

7. (Opcional) Si desea exigir que las aprobaciones de una solicitud de extracción procedan de un grupo específico de usuarios, en Approval rule members (Miembros de regla de aprobación), elija Add (Añadir). En Approver type (Tipo de aprobador), elija una de las opciones siguientes:
 - Nombre de usuario de IAM o rol asumido: esta opción rellena previamente el ID de cuenta de AWS de la cuenta que utilizó para iniciar sesión y solo requiere un nombre. Se puede utilizar tanto para usuarios de IAM como para usuarios de acceso federado cuyo nombre coincida con el nombre proporcionado. Se trata de una opción muy potente que ofrece una gran flexibilidad. Por ejemplo, si ha iniciado sesión con la cuenta de Amazon Web Services 123456789012 y especifica esta opción, así como **Mary_Major**, todas las opciones siguientes se contabilizarán como aprobaciones procedentes de ese usuario:
 - Un usuario de IAM en la cuenta (arn:aws:iam::123456789012:user/Mary_Major)

- Un usuario federado identificado en IAM como Mary_Major
(arn:aws:sts::123456789012:federated-user/Mary_Major)

Esta opción no reconocerá una sesión activa de alguien que asuma el rol **CodeCommitReview** con el nombre de sesión de rol Mary_Major (arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major) a menos que incluya un comodín (*Mary_Major). También puede especificar el nombre del rol explícitamente (CodeCommitReview/Mary_Major).

- ARN completo: esta opción le permite especificar el nombre de recurso de Amazon (ARN) completo del usuario de IAM o rol. Esta opción también admite las funciones asumidas que utilizan otros AWS servicios, como AWS Lambda y AWS CodeBuild. Para los roles asumidos, el formato de ARN debe ser arn:aws:sts::*AccountID*:assumed-role/*RoleName* para los roles y arn:aws:sts::*AccountID*:assumed-role/*FunctionName* para las funciones.

Si eligió el nombre de usuario de IAM o el rol asumido como tipo de aprobador, en Valor, introduzca el nombre del usuario de IAM o rol o el ARN completo del usuario o rol. Elija Add (Agregar) de nuevo para agregar más usuarios o roles, hasta que haya agregado todos los usuarios o roles cuyas aprobaciones se contabilizan en el número de aprobaciones necesarias.

Ambos tipos de aprobadores le permiten utilizar comodines (*) en los valores. Por ejemplo, si elige la opción de nombre de usuario de IAM o rol asumido y especifica **CodeCommitReview/***, todos los usuarios que asumen el rol de **CodeCommitReview** se cuentan en el grupo de aprobación. Sus nombres de sesión de rol individuales se contabilizan en el número necesario de aprobadores. De esta manera, tanto Mary_Major como Li_Juan se contabilizan como aprobaciones cuando se inicia sesión y se asume el rol de CodeCommitReview. Para obtener más información sobre IAM ARNs, los caracteres comodín y los formatos, consulte Identificadores de [IAM](#).

 Note

Las reglas de aprobación no admiten aprobaciones entre cuentas.

8. Cuando haya terminado de configurar la regla de aprobación, elija Submit (Enviar).

Crear una regla de aprobación para una solicitud de extracción (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el. AWS CLI Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para crear una regla de aprobación para una solicitud de extracción en un CodeCommit repositorio

- Ejecute el comando `create-pull-request-approval-rule`, especificando:
 - El ID de la solicitud de extracción (con la opción `--id`).
 - El nombre de la regla de aprobación (con la opción `--approval-rule-name`).
 - El contenido de la regla de aprobación (con la opción `--approval-rule-content`).

Al crear la regla de aprobación, puede especificar a los aprobadores de un grupo de aprobación de dos maneras:

- **CodeCommitApprovers**: Esta opción solo requiere una cuenta de Amazon Web Services y un recurso. Se puede utilizar tanto para usuarios de IAM como para usuarios de acceso federado cuyo nombre coincida con el nombre de recurso proporcionado. Se trata de una opción muy potente que ofrece una gran flexibilidad. Por ejemplo, si especifica la cuenta 123456789012 de Amazon Web Services y **Mary_Major**, todas las opciones siguientes se contabilizan como aprobaciones procedentes de ese usuario:
 - Un usuario de IAM en la cuenta (`arn:aws:iam::123456789012:user/Mary_Major`)
 - Un usuario federado identificado en IAM como `Mary_Major`
(`arn:aws:sts::123456789012:federated-user/Mary_Major`)

Esta opción no reconocerá una sesión activa de alguien que asuma el rol **CodeCommitReview** con el nombre de sesión de rol `Mary_Major` (`arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major`) a menos que incluya un comodín (`*Mary_Major`).

- **ARN completo**: esta opción le permite especificar el nombre de recurso de Amazon (ARN) completo del usuario de IAM o rol.

Para obtener más información sobre la IAM ARNs, los caracteres comodín y los formatos, consulte [Identificadores de IAM](#).

En el ejemplo siguiente se crea una regla de aprobación denominada `Require two approved approvers` para una solicitud de extracción con el ID 27. La regla especifica que se requieren dos aprobaciones de un grupo de aprobación. El grupo incluye a todos los usuarios que acceden a la cuenta de 123456789012 Amazon Web Services CodeCommit y asumen su función **CodeCommitReview** en ella. También incluye un usuario de IAM o un usuario federado denominado `Nikhil_Jayashankar` en la misma cuenta de Amazon Web Services:

```
aws codecommit create-pull-request-approval-rule --pull-request-id 27
--approval-rule-name "Require two approved approvers" --approval-
rule-content "{\"Version\": \"2018-11-08\", \"Statements\": [{\"Type\":
  \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
  \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Ver solicitudes de cambios en un AWS CodeCommit repositorio

Puedes usar la AWS CodeCommit consola o la AWS CLI para ver las solicitudes de extracción de tu repositorio. De forma predeterminada, solo se ven las solicitudes de extracción abiertas, pero puede cambiar el filtro para ver todas las solicitudes de extracción, solo las solicitudes cerradas, solo las solicitudes de extracción que ha creado, etc.

Temas

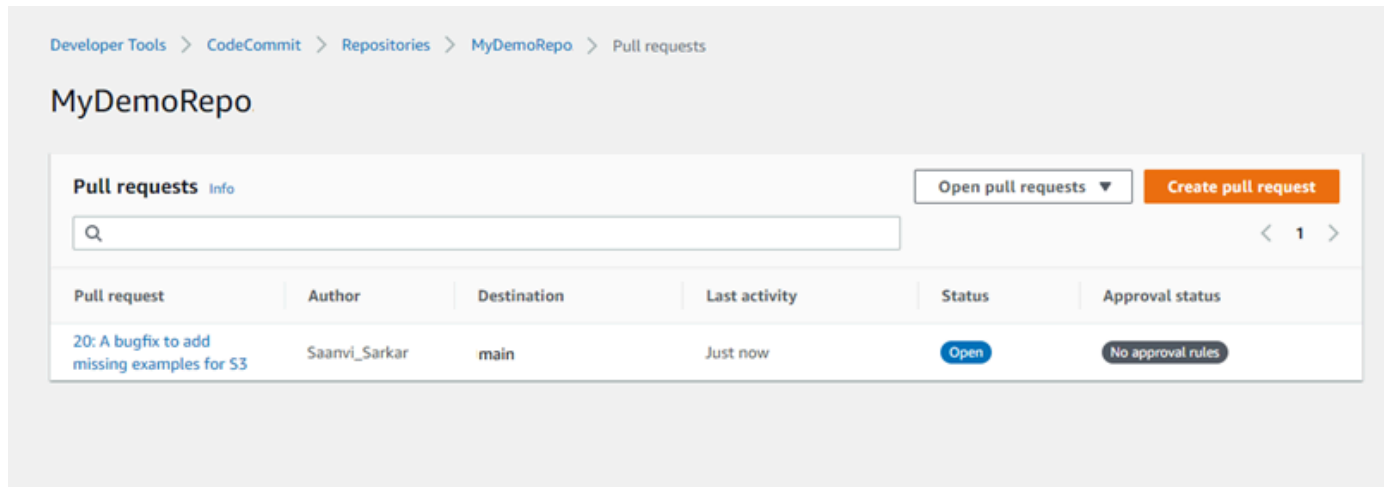
- [Visualizar solicitudes de extracción \(consola\)](#)
- [Visualizar solicitudes de extracción \(AWS CLI\)](#)

Visualizar solicitudes de extracción (consola)

Puedes usar la AWS CodeCommit consola para ver una lista de solicitudes de incorporación de cambios en un CodeCommit repositorio. Al cambiar el filtro, puede cambiar la visualización de la lista para que solo muestre un conjunto determinado de solicitudes de extracción. Por ejemplo, puede ver una lista de las solicitudes de extracción que ha creado cuyo estado es `Open` o puede elegir otro filtro y ver las solicitudes de extracción que ha creado cuyo estado es `Closed`.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas solicitudes de extracción desea ver.

3. En el panel de navegación, elija Pull Requests.
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas.



5. Para cambiar el filtro de visualización, elija en la lista de filtros disponibles:
 - Open pull requests (Solicitudes de extracción abiertas) (opción predeterminada): muestra todas las solicitudes de extracción cuyo estado es Open (Abierto).
 - All pull requests (Todas las solicitudes de extracción): muestra todas las solicitudes de extracción.
 - Closed pull requests (Solicitudes de extracción cerradas): muestra todas las solicitudes de extracción cuyo estado es Closed (Cerrado).
 - My pull requests (Mis solicitudes de extracción): muestra todas las solicitudes de extracción que ha creado, sea cual sea su estado. No muestra las revisiones que ha comentado o en las que ha participado.
 - My open pull requests (Mis solicitudes de extracción abiertas): muestra todas las solicitudes de extracción que ha creado cuyo estado es Open (Abierta).
 - My closed pull requests (Mis solicitudes de extracción cerradas): muestra todas las solicitudes de extracción que ha creado cuyo estado es Closed (Cerrada).
6. Cuando encuentre una solicitud de extracción en la lista mostrada que le gustaría ver, elíjala.

Visualizar solicitudes de extracción (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Sigue estos pasos para utilizarla AWS CLI para ver las solicitudes de incorporación de cambios en un CodeCommit repositorio.

1. Para ver una lista de las solicitudes de extracción en un repositorio, ejecute el comando `list-pull-requests`, especificando:
 - El nombre del CodeCommit repositorio en el que quieres ver las solicitudes de extracción (con la `--repository-name` opción).
 - (Opcional) El estado de la solicitud de extracción (con la opción `--pull-request-status`).
 - (Opcional) El nombre de recurso de Amazon (ARN) del usuario de IAM que ha creado la solicitud de extracción (con la opción `--author-arn`).
 - (Opcional) Un token de enumeración que se puede utilizar para devolver lotes de resultados (con la opción `--next-token`).
 - (Opcional) Un límite en el número de resultados devueltos por solicitud (con la opción `--max-results`).

Por ejemplo, para enumerar las solicitudes de extracción creadas por un usuario de IAM con el `arn:aws:iam::111111111111:user/Li_Juan` ARN y el estado `CLOSED` de en CodeCommit un repositorio denominado: `MyDemoRepo`

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "nextToken": "",
  "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

IDs Las solicitudes de extracción se muestran en el orden de la actividad más reciente.

2. Para ver los detalles de una solicitud de extracción, ejecute el comando `get-pull-request` con la opción `--pull-request-id`, especificando el ID de la solicitud de extracción. Por ejemplo, para ver información sobre una solicitud de extracción con el identificador de `27`:

```
aws codecommit get-pull-request --pull-request-id 27
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 1571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "lastActivityDate": 1562619583.565,
    "pullRequestTargets": [
      {
        "sourceCommit": "ca45e279EXAMPLE",
        "sourceReference": "refs/heads/bugfix-1234",
        "mergeBase": "a99f5ddbEXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": false
        },
        "destinationCommit": "2abfc6beEXAMPLE",
        "repositoryName": "MyDemoRepo"
      }
    ],
    "revisionId": "e47def21EXAMPLE",
    "title": "Quick fix for bug 1234",
    "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
    "clientRequestToken": "d8d7612e-EXAMPLE",
    "creationDate": 1562619583.565,
    "pullRequestId": "27",
    "pullRequestStatus": "OPEN"
  }
}
```

3.

Para ver las aprobaciones en una solicitud de extracción, ejecute el comando `get-pull-request-approval-state` especificando lo siguiente:

- El ID de la solicitud de extracción (con la opción `--pull-request-id`).
- El ID de revisión de la solicitud de extracción (mediante `--revision-id` option)). Puede obtener el ID de revisión actual de una solicitud de extracción mediante el comando [get-pull-request](#).

Por ejemplo, para ver las aprobaciones de una solicitud de extracción con un ID `8` y un ID de revisión de `9f29d167EXAMPLE`:

```
aws codecommit get-pull-request-approval-state --pull-request-id 8 --revision-id 9f29d167EXAMPLE
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "approvals": [
    {
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
      "approvalState": "APPROVE"
    }
  ]
}
```

4. Para ver los eventos de una solicitud de extracción, ejecute el comando `describe-pull-request-events` con la opción `--pull-request-id`, especificando el ID de la solicitud de extracción. Por ejemplo, para ver los eventos de una solicitud de extracción con el ID de `8`:

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "pullRequestEvents": [
    {
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_CREATED",
      "eventDate": 1510341779.53,
      "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
    }
  ]
}
```



```

    },
    {
      "pullRequestStatusChangedEventMetadata": {
        "pullRequestStatus": "CLOSED"
      },
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
      "eventDate": 1510341930.72,
      "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
    }
  ]
}

```

5. Para ver si existen conflictos de combinación para una solicitud de extracción, ejecute el comando `get-merge-conflicts`, especificando:
 - El nombre del CodeCommit repositorio (con la `--repository-name` opción).
 - La ramificación, etiqueta, HEAD u otro tipo de referencia completa para el origen de los cambios que se va a utilizar en la evaluación de combinación (con la opción `--source-commit-specifier`).
 - La ramificación, etiqueta, HEAD u otro tipo de referencia completa para el destino de los cambios que se va a utilizar en la evaluación de combinación (con la opción `--destination-commit-specifier`).
 - La opción de combinación que se va a utilizar (con la opción `--merge-option`).

Por ejemplo, para ver si hay algún conflicto de fusión entre la punta de una rama de origen denominada *my-feature-branch* y una rama de destino denominada *main* en un repositorio denominado *MyDemoRepo*:

```

aws codecommit get-merge-conflicts --repository-name MyDemoRepo --source-commit-specifier my-feature-branch --destination-commit-specifier main --merge-option FAST_FORWARD_MERGE

```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```

{
  "destinationCommitId": "fac04518EXAMPLE",
  "mergeable": false,
  "sourceCommitId": "16d097f03EXAMPLE"
}

```

```
}
```

Revisión de una solicitud de extracción

Puedes usar la AWS CodeCommit consola para revisar los cambios incluidos en una solicitud de extracción. Puede añadir comentarios a la solicitud, a los archivos y a líneas de código individuales. También puede responder a los comentarios de otros usuarios. Si el repositorio está [configurado con notificaciones](#), recibirá un correo electrónico cuando los usuarios respondan a sus comentarios o cuando hagan comentarios sobre una solicitud de extracción.

Puedes utilizarla AWS CLI para comentar una solicitud de cambios y responder a los comentarios. Para revisar los cambios, debes usar la CodeCommit consola, el git diff comando o una herramienta de diferenciación.

Temas

- [Revisión de una solicitud de extracción \(consola\)](#)
- [Revisar solicitudes de extracción \(AWS CLI\)](#)

Revisión de una solicitud de extracción (consola)

Puedes usar la CodeCommit consola para revisar una solicitud de incorporación de cambios en un CodeCommit repositorio.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione el nombre del repositorio.
3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas. Elija la solicitud de extracción abierta que desea revisar.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests

MyDemoRepo

Pull requests [Info](#)

< 1 >

[Open pull requests](#) [Create pull request](#)

Pull request	Author	Destination	Last activity	Status	Approval status
20: A bugfix to add missing examples for S3	Saanvi_Sarkar	main	Just now	Open	No approval rules

Note

Puede comentar una solicitud de extracción cerrada o combinada, pero no puede combinarla ni volver a abrirla.

- En la solicitud de extracción, elija **Changes**.
- Realice una de las siguientes acciones:
 - Para añadir un comentario general para la solicitud de extracción en su conjunto, en **Comments on changes** (Comentarios de los cambios), bajo **New comment** (Nuevo comentario), introduzca un comentario y, a continuación, seleccione **Save** (Guardar). Puede utilizar [Markdown \(Marcado\)](#) o introducir un comentario en texto sin formato.

Comments on changes

New comment ☐ Preview markdown [Learn more](#)

Did we also change the variable name in blf.py and concat.py?

[Save](#)

- Para añadir un comentario a un archivo en la confirmación, en **Changes**, busque el nombre del archivo. Elija el icono de comentario



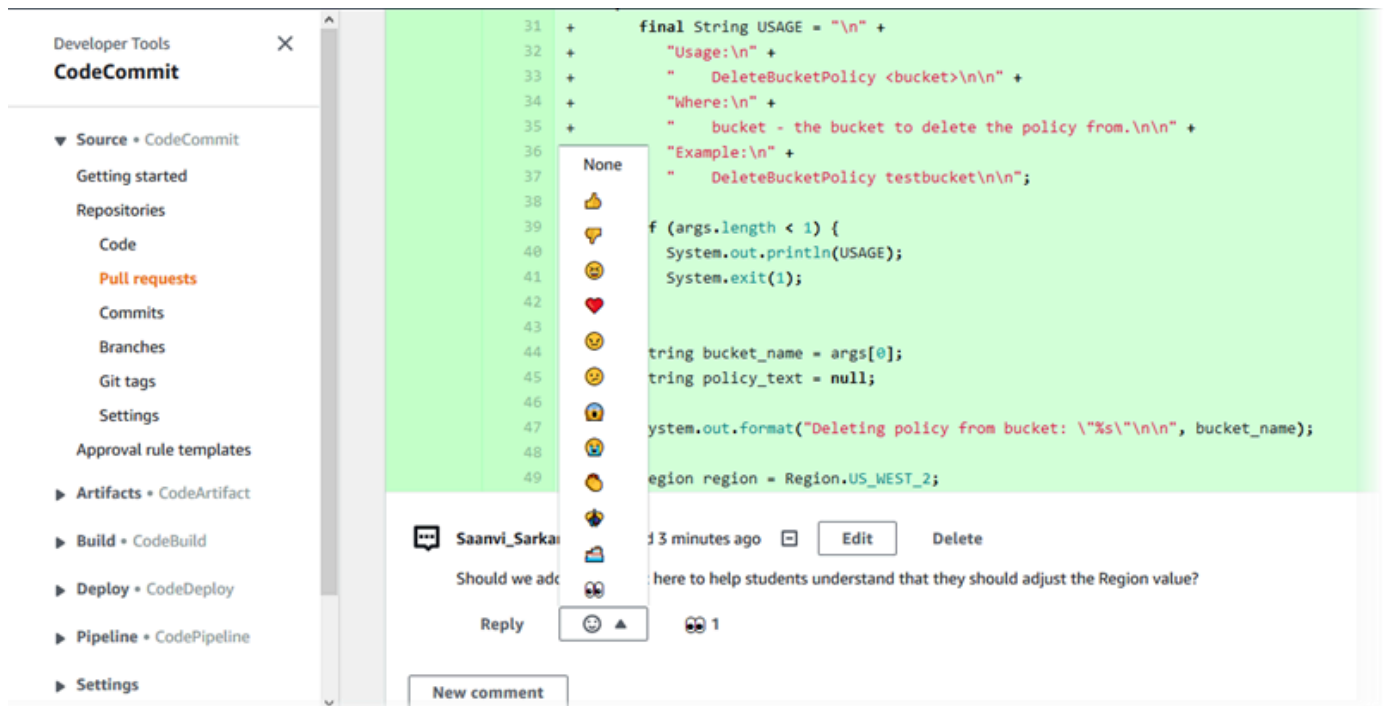
que aparece junto al nombre de archivo, introduzca un comentario y, a continuación, elija Save (Guardar).

- Para añadir un comentario a una línea modificada en la solicitud de extracción, en Changes, vaya a la línea sobre la que desea realizar un comentario. Elija el icono de comentario



que aparece para esa línea, introduzca un comentario y, a continuación, elija Save (Guardar).

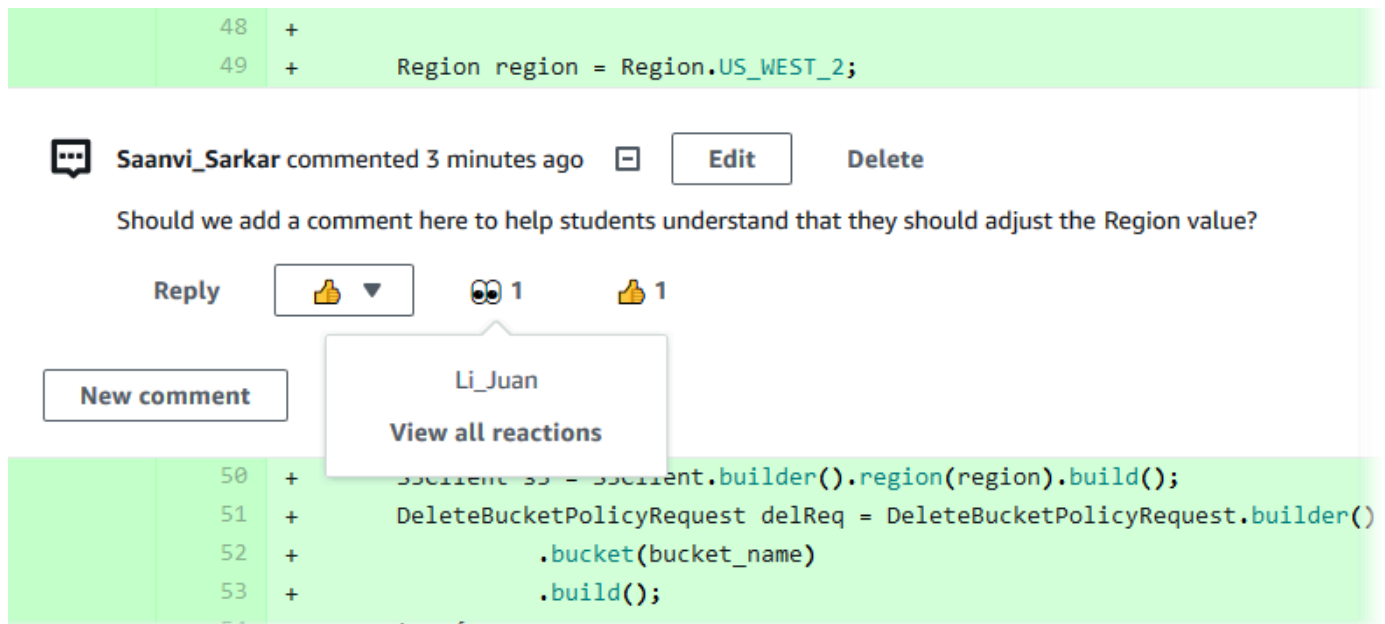
7. Para responder a los comentarios sobre una confirmación, en Changes o en Activity, elija Reply. Puede responder con texto y emojis.



Para ver los nombres de las personas que respondieron con una respuesta de reacción con emoji en particular, selecciónela. Para ver todas las reacciones con emojis e información sobre quién respondió y con qué emojis lo hizo, seleccione Ver todas las reacciones. Si respondió con un emoji a un comentario, su respuesta se mostrará en el icono del botón de reacción con emoji.

Note

Los recuentos de reacciones que se muestran en la consola son exactos en el momento en que se cargó la página. Para obtener la información más actualizada sobre el recuento de reacciones de los emojis, actualice la página o seleccione Ver todas las reacciones.



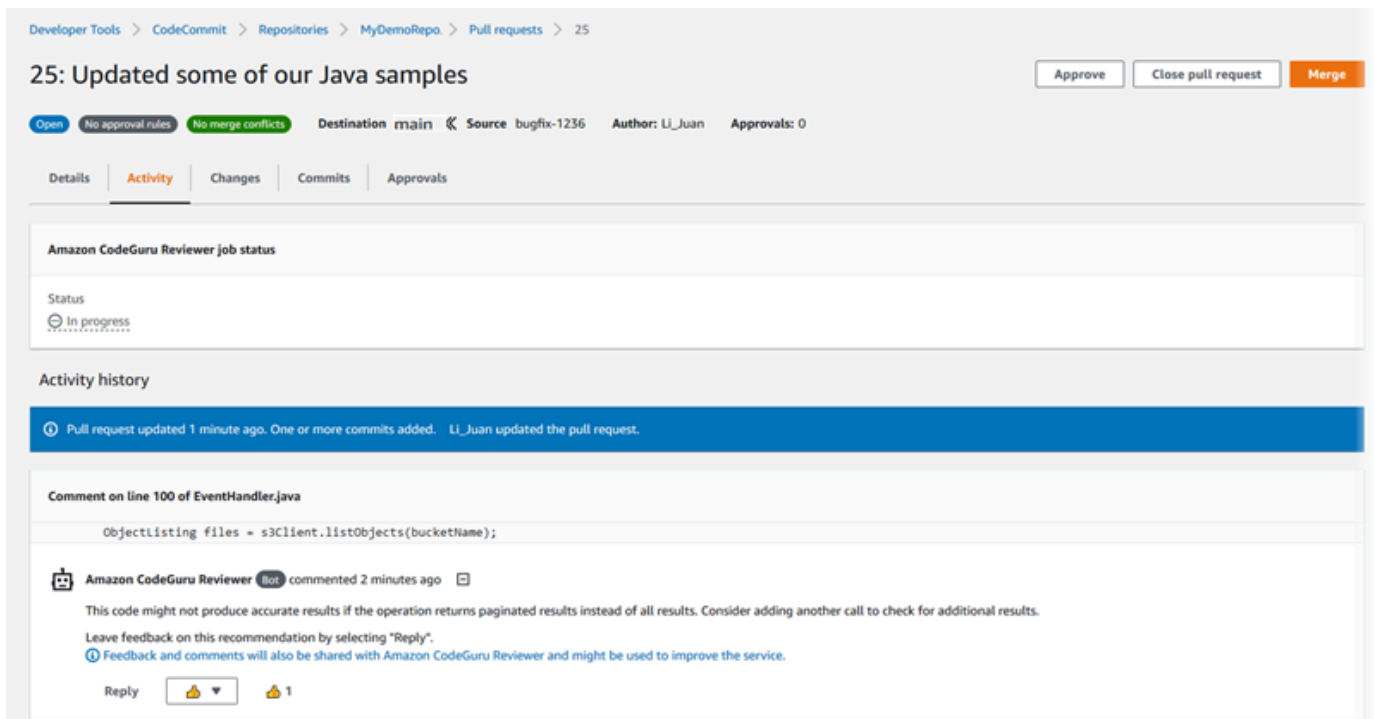
8. (Opcional) Para responder a una recomendación creada por Amazon CodeGuru Reviewer, incluida la aportación de comentarios sobre la calidad de la recomendación, selecciona Responder. Utilice los botones de reacción para proporcionar información general acerca de si aprueba o no la recomendación. Utilice el campo de comentario para proporcionar más detalles sobre su reacción.

Note

Amazon CodeGuru Reviewer es un servicio de revisión de código automatizado que utiliza el análisis de programas y el aprendizaje automático para detectar problemas comunes y recomendar soluciones en el código de Java o Python.

- Solo verás los comentarios de Amazon CodeGuru Reviewer si has asociado el repositorio a Amazon CodeGuru Reviewer, si el análisis está completo y si el código de la solicitud de extracción es código Java o Python. Para obtener más información, consulte [Amazon CodeGuru Reviewer y AWS CodeCommit](#).
- Los comentarios de Amazon CodeGuru Reviewer solo aparecen en la pestaña Cambios si se hicieron en la revisión más reciente de la solicitud de cambios. Siempre aparecen en la pestaña Actividad.
- Si bien puedes responder con cualquiera de las reacciones de emoji disponibles a las recomendaciones de Amazon CodeGuru Reviewer, solo se utilizan las reacciones de

emoji con el pulgar hacia arriba y el pulgar hacia abajo para evaluar la utilidad de la recomendación.



9. Para aprobar los cambios realizados en una solicitud de extracción, elija Approve (Aprobar).

Note

No puede aprobar una solicitud de extracción que ha creado.

Puede ver aprobaciones, reglas de aprobación para una solicitud de extracción y reglas de aprobación creadas por plantillas de reglas de aprobación en Approvals (Aprobaciones). Si decide no aprobar la solicitud de extracción después de todo, puede elegir Revoke approval (Revocar aprobación).

Note

Solo puede aprobar o revocar la aprobación en una solicitud de extracción abierta. No se puede aprobar ni revocar la aprobación de una solicitud de extracción cuyo estado sea Merged (Combinada) o Closed (Cerrada).

The screenshot shows the AWS CodeCommit pull request interface for a pull request titled "14: Hotfix for customer issue 1234". The interface includes a breadcrumb trail: Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 14. At the top right, there are buttons for "Revoke approval", "Close pull request", and "Merge". Below the title, there are status indicators: "Open", "Approved", and "Mergeable". The "Destination" is set to "master" and the "Source" is "Source". The "Author" is "Author:" and the "Approvals" count is "2". The "Approvals" tab is selected, showing a table of approvers and their status. Below the table, there is an "Approval rules" section with a table of rules and their status.

Approver	Status
Li_Juan	Approved
Saanvi_Sarkar	Approved

Approval rule	Status
My Approval Rule	Rule satisfied

Revisar solicitudes de extracción (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Puedes revisar las solicitudes de cambios con los siguientes AWS CLI comandos:

- [post-comment-for-pull-request](#) para añadir un comentario a una solicitud de extracción
- [get-comments-for-pull-request](#) para ver los comentarios en una solicitud de extracción
- [update-pull-request-approval-state](#) para aprobar o revocar la aprobación de una solicitud de extracción
- [post-comment-reply](#) para responder a un comentario en una solicitud de extracción

También puede usar emojis con comentarios en una solicitud de extracción con los siguientes comandos:

- Para responder a un comentario con un emoji, ejecute [put-comment-reaction](#).
- Para ver las reacciones con emojis a un comentario, ejecute [get-comment-reactions](#).

Para usar el AWS CLI para revisar las solicitudes de extracción en un CodeCommit repositorio

1. Para añadir un comentario a una solicitud de extracción, ejecute el comando `post-comment-for-pull-request` especificando lo siguiente:
 - El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El nombre del repositorio que contiene la solicitud de extracción (con la opción `--repository-name`).
 - El ID completo de la confirmación en la ramificación de destino en la que se combina la solicitud de extracción (con la opción `--before-commit-id`).
 - El ID completo de la confirmación de la ramificación de origen que es el extremo actual de la ramificación de la solicitud de extracción en el momento en que se publica el comentario (con la opción `--after-commit-id`).
 - Un token de idempotencia único generado por el cliente (con la opción `--client-request-token`).
 - El contenido del comentario (con la opción `--content`).
 - Una lista con información de la ubicación sobre dónde se puede introducir el comentario, que incluya:
 - El nombre del archivo que se está comparando, incluida la extensión y el subdirectorío, si los hubiese (con el atributo `filePath`).
 - El número de línea del cambio del archivo comparado (con el atributo `filePosition`).
 - Si el comentario sobre el cambio se ha realizado “antes” o “después” en la comparación entre la ramificación de origen y la de destino (con el atributo `relativeFileVersion`).

Por ejemplo, usa este comando para añadir el comentario *"These don't appear to be used anywhere. Can we remove them?"* sobre el cambio al `ahs_count.py` archivo de una solicitud de extracción con el ID *47* de un repositorio denominado *MyDemoRepo*.

```
aws codecommit post-comment-for-pull-request --pull-request-id "47" --
repository-name MyDemoRepo --before-commit-id 317f8570EXAMPLE --after-
commit-id 5d036259EXAMPLE --client-request-token 123Example --content
"These don't appear to be used anywhere. Can we remove them?" --location
filePath=ahs_count.py,filePosition=367,relativeFileVersion=AFTER
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente.

```
{
```

```

    "afterBlobId": "1f330709EXAMPLE",
    "afterCommitId": "5d036259EXAMPLE",
    "beforeBlobId": "80906a4cEXAMPLE",
    "beforeCommitId": "317f8570EXAMPLE",
    "comment": {
      "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
      "clientRequestToken": "123Example",
      "commentId": "abcd1234EXAMPLEb5678efgh",
      "content": "These don't appear to be used anywhere. Can we remove
them?",
      "creationDate": 1508369622.123,
      "deleted": false,
      "lastModifiedDate": 1508369622.123,
      "callerReactions": [],
      "reactionCounts": []
    },
    "location": {
      "filePath": "ahs_count.py",
      "filePosition": 367,
      "relativeFileVersion": "AFTER"
    },
    "repositoryName": "MyDemoRepo",
    "pullRequestId": "47"
  }
}

```

2. Para ver los comentarios de una solicitud de extracción, ejecute el comando `get-comments-for-pull-request` especificando lo siguiente:
 - El nombre del CodeCommit repositorio (con la `--repository-name` opción).
 - El ID generado por el sistema de la solicitud de extracción (con la opción `--pull-request-id`).
 - (Opcional) Un token de enumeración para devolver el siguiente lote de resultados (con la opción `--next-token`).
 - (Opcional) Un número entero no negativo para limitar en el número de resultados devueltos (con la opción `--max-results`).

Por ejemplo, utilice este comando para ver los comentarios de una solicitud de extracción con el ID 42.

```
aws codecommit get-comments-for-pull-request --pull-request-id 42
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente.

```
{
  "commentsForPullRequestData": [
    {
      "afterBlobId": "1f330709EXAMPLE",
      "afterCommitId": "5d036259EXAMPLE",
      "beforeBlobId": "80906a4cEXAMPLE",
      "beforeCommitId": "317f8570EXAMPLE",
      "comments": [
        {
          "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
          "clientRequestToken": "",
          "commentId": "abcd1234EXAMPLEb5678efgh",
          "content": "These don't appear to be used anywhere. Can we remove
them?",
          "creationDate": 1508369622.123,
          "deleted": false,
          "lastModifiedDate": 1508369622.123,
          "callerReactions": [],
          "reactionCounts":
            {
              "THUMBSUP" : 6,
              "CONFUSED" : 1
            }
        },
        {
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
          "clientRequestToken": "",
          "commentId": "442b498bEXAMPLE5756813",
          "content": "Good catch. I'll remove them.",
          "creationDate": 1508369829.104,
          "deleted": false,
          "lastModifiedDate": 150836912.273,
          "callerReactions": ["THUMBSUP"]
          "reactionCounts":
            {
              "THUMBSUP" : 14
            }
        }
      ],
      "location": {
        "filePath": "ahs_count.py",
```

```
        "filePosition": 367,  
        "relativeFileVersion": "AFTER"  
    },  
    "repositoryName": "MyDemoRepo",  
    "pullRequestId": "42"  
  }  
],  
"nextToken": "exampleToken"  
}
```

3.

Para aprobar o revocar la aprobación de una solicitud de extracción, ejecute el comando `update-pull-request-approval-state` especificando lo siguiente:

- El ID de la solicitud de extracción (con la opción `--pull-request-id`).
- El ID de revisión de la solicitud de extracción (mediante `--revision-id` option)). Puede obtener el ID de revisión actual de una solicitud de extracción mediante el comando [get-pull-request](#).
- El estado de aprobación que desea aplicar (mediante la opción `--approval-state`). Los estados de aprobación válidos incluyen `APPROVE` y `REVOKE`.

Por ejemplo, usa este comando para aprobar una solicitud de extracción con el identificador **27** y el identificador de revisión de **9f29d167EXAMPLE**.

```
aws codecommit update-pull-request-approval-state --pull-request-id 27 --revision-id 9f29d167EXAMPLE --approval-state "APPROVE"
```

Si se ejecuta correctamente, este comando no devuelve nada.

4. Para publicar una respuesta a un comentario en una solicitud de extracción, ejecute el comando `post-comment-reply` especificando lo siguiente:

- El ID generado por el sistema del comentario al que desea responder (con la opción `--in-reply-to`).
- Un token de idempotencia único generado por el cliente (con la opción `--client-request-token`).
- El contenido de la respuesta (con la opción `--content`).

Por ejemplo, usa este comando para añadir la respuesta **"Good catch. I'll remove them."** al comentario con el identificador generado por el sistema de **abcd1234EXAMPLEb5678efgh**

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --  
content "Good catch. I'll remove them." --client-request-token 123Example
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente.

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "123Example",  
    "commentId": "442b498bEXAMPLE5756813",  
    "content": "Good catch. I'll remove them.",  
    "creationDate": 1508369829.136,  
    "deleted": false,  
    "lastModifiedDate": 150836912.221,  
    "callerReactions": [],  
    "reactionCounts": []  
  }  
}
```

Actualización de una solicitud de extracción

Puede actualizar una solicitud de cambios de código con más cambios de código enviando las confirmaciones a la ramificación fuente de una solicitud de extracción abierta. Para obtener más información, consulte [Crea una confirmación en AWS CodeCommit](#).

Puede utilizar la consola AWS CodeCommit o la AWS CLI para actualizar el título o la descripción de una solicitud de extracción. Es posible que desee actualizar la solicitud de extracción porque:

- Otros usuarios no entienden la descripción, o el título original es confuso.
- Desea que el título o la descripción reflejen los cambios realizados en la ramificación de origen de una solicitud de extracción abierta.

Actualizar una solicitud de extracción (consola)

Puede utilizar la consola de CodeCommit para actualizar el título y la descripción de una solicitud de extracción en un repositorio de CodeCommit. Para actualizar el código de la solicitud de extracción, envíe las confirmaciones a la ramificación fuente de una solicitud de extracción abierta.

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuya solicitud de extracción desea actualizar.
3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas. Elija la solicitud de extracción abierta que desea actualizar.
5. En la solicitud de extracción, elija Details (Detalles) y, a continuación, seleccione Edit details (Editar detalles) para editar el título o la descripción.

 Note

No puede actualizar el título o la descripción de una solicitud de extracción cerrada o combinada.

Actualizar solicitudes de extracción (AWS CLI)

Para utilizar los comandos de AWS CLI en CodeCommit, instale la AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

También pueden interesarle los siguientes comandos:

- [update-pull-request-approval-state](#), para aprobar o revocar la aprobación en una solicitud de extracción.
- [create-pull-request-approval-rule](#), para crear una regla de aprobación para una solicitud de extracción.
- [delete-pull-request-approval-rule](#), para eliminar una regla de aprobación para una solicitud de extracción.
- [Crea una confirmación mediante el AWS CLI](#) o [Creación de una confirmación mediante un cliente de Git](#) para crear e insertar cambios de código adicionales en la ramificación fuente de una solicitud de extracción abierta.

Para usar la AWS CLI para actualizar solicitudes de extracción en un repositorio de CodeCommit

1. Para actualizar el título de una solicitud de extracción en un repositorio, ejecute el comando `update-pull-request-title`, especificando:

- El ID de la solicitud de extracción (con la opción `--pull-request-id`).
- El título de la solicitud de extracción (con la opción `--title`).

Por ejemplo, para actualizar el título de una solicitud de extracción con el ID **47**:

```
aws codecommit update-pull-request-title --pull-request-id 47 --title  
"Consolidation of global variables - updated review"
```

2. Para actualizar la descripción de una solicitud de extracción, ejecute el comando `update-pull-request-description` especificando:

- El ID de la solicitud de extracción (con la opción `--pull-request-id`).
- La descripción (con la opción `--description`).

Por ejemplo, para actualizar la descripción de una solicitud de extracción con el ID **47** :

```
aws codecommit update-pull-request-description --pull-request-id 47 --description  
"Updated the pull request to remove unused global variable."
```

Editar o eliminar una regla de aprobación para una solicitud de extracción

Cuando hay una regla de aprobación en una solicitud de extracción, no se puede combinar hasta que se hayan cumplido sus condiciones. Puede cambiar las reglas de aprobación de las solicitudes de extracción para facilitar el cumplimiento de sus condiciones o para aumentar el rigor de las revisiones. Puede cambiar el número de usuarios que deben aprobar una solicitud de extracción. También puede añadir, quitar o cambiar la pertenencia a un grupo de usuarios de aprobación para la regla. Por último, si ya no desea utilizar una regla de aprobación para una solicitud de extracción, puede eliminarla.

Note

También puede sustituir las reglas de aprobación para una solicitud de extracción. Para obtener más información, consulte [Anular reglas de aprobación en una solicitud de extracción](#).

Puede usar la AWS CodeCommit consola o la AWS CLI para editar y eliminar las reglas de aprobación de su repositorio.

Temas

- [Editar o eliminar una regla de aprobación para una solicitud de extracción \(consola\)](#)
- [Editar o eliminar una regla de aprobación para una solicitud de extracción \(AWS CLI\)](#)

Editar o eliminar una regla de aprobación para una solicitud de extracción (consola)

Puedes usar la CodeCommit consola para editar o eliminar una regla de aprobación para una solicitud de extracción en un CodeCommit repositorio.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio donde desea crear o editar una regla de aprobación para una solicitud de extracción.
3. En el panel de navegación, elija Pull Requests.
4. Elija la solicitud de extracción en la que desee editar o eliminar una regla de aprobación. Solo puede editar y eliminar reglas de aprobación para las solicitudes de extracción abiertas.

5. En la solicitud de extracción, elija Approvals (Aprobaciones) y, a continuación, elija la regla que desea editar o eliminar de la lista. Realice una de las siguientes acciones:
 - Si desea editar la regla, seleccione Edit (Editar).
 - Si desea eliminar la regla, elija Delete (Eliminar) y, a continuación, siga las instrucciones para verificar la eliminación de la regla.
6. En Edit approval rule (Editar regla de aprobación), realice los cambios que desee en la regla y, a continuación, elija Submit (Enviar).

Edit approval rule

Rule details

Rule name
My Approval Rule

Number of approvals needed
1

Approval pool members - *optional*
If approval pool members are specified, only approvals from these members will count toward satisfying this rule. Use a wildcard to match multiple approvers with one value.

Approver type Info	Value	
CodeCommit ▼	Mary_Major	Remove
CodeCommit ▼	Li_Juan	Remove
CodeCommit ▼	Saanvi_Sarkar	Remove
Fully qualified ARN ▼	arn:aws:iam:::user/	Remove

Add

Cancel **Submit**

7. Cuando haya terminado de configurar la regla de aprobación, elija Submit (Enviar).

Editar o eliminar una regla de aprobación para una solicitud de extracción (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instala el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Puede utilizar el AWS CLI para editar el contenido de una regla de aprobación y eliminar una regla de aprobación.

Note

También pueden interesarle los siguientes comandos:

- [update-pull-request-approval-state](#), para aprobar o revocar la aprobación en una solicitud de extracción.
- [get-pull-request-approval-states](#), para ver las aprobaciones en la solicitud de extracción.
- [evaluate-pull-request-approval-rules](#), para determinar si se han cumplido las condiciones de las normas de aprobación para una solicitud de extracción.

Para usar la AWS CLI para editar o eliminar una regla de aprobación para una solicitud de extracción en un CodeCommit repositorio

1. Para editar una regla de aprobación, ejecute el comando `update-pull-request-approval-rule-content` especificando lo siguiente:
 - El ID de la solicitud de extracción (con la opción `--id`).
 - El nombre de la regla de aprobación (con la opción `--approval-rule-name`).
 - El contenido de la regla de aprobación (con la opción `--approval-rule-content`).

En este ejemplo, se actualiza una regla de aprobación denominada *Require two approved approvers* para una solicitud de extracción con el ID de `27`. La regla requiere la aprobación de un usuario de un grupo de aprobaciones que incluya a todos los usuarios de IAM de la cuenta de `123456789012` Amazon Web Services:

```
aws codecommit update-pull-request-approval-rule-content --pull-request-id 27
--approval-rule-name "Require two approved approvers" --approval-rule-content
```

```
"{Version: 2018-11-08, Statements: [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:[\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

2.

Para eliminar una regla de aprobación, ejecute el comando `delete-pull-request-approval-rule` especificando lo siguiente:

- El ID de la solicitud de extracción (con la opción `--id`).
- El nombre de la regla de aprobación (con la opción `--approval-rule-name`).

Por ejemplo, para eliminar una regla de aprobación con el nombre de una *My Approval Rule* solicitud de extracción con el identificador de *15*:

```
aws codecommit delete-pull-request-approval-rule --pull-request-id 15 --approval-rule-name "My Approval Rule"
```

Anular reglas de aprobación en una solicitud de extracción

Durante un proceso normal de desarrollo, es conveniente que los usuarios cumplan las condiciones de las reglas de aprobación antes de combinar las solicitudes de extracción. Sin embargo, puede haber ocasiones en las que necesite acelerar la combinación de una solicitud de extracción. Por ejemplo, es posible que desee llevar al entorno de producción una corrección de errores, pero que no haya nadie disponible en el grupo de aprobación para aprobar la solicitud de extracción. En casos como estos, puede optar por anular las reglas de aprobación en una solicitud de extracción. Puede sustituir todas las reglas de aprobación de una solicitud de extracción, incluidas las creadas específicamente para la solicitud de extracción y generadas a partir de una plantilla de regla de aprobación. No puede anular selectivamente una regla de aprobación específica, sino solamente todas ellas. Una vez que haya dejado a un lado los requisitos de regla de aprobación mediante la anulación de las reglas, puede combinar la solicitud de extracción con su ramificación de destino.

Cuando se anulan las reglas de aprobación de una solicitud de extracción, la información sobre el usuario que ha anulado las reglas se registra en la actividad de la solicitud de extracción. De esta manera, puede volver al historial de una solicitud de extracción y revisar quién anuló las reglas. También puede optar por revocar la anulación si la solicitud de extracción sigue abierta. Después de haber combinado la solicitud de extracción, ya no se puede revocar su anulación.

Temas

- [Anular reglas de aprobación \(consola\)](#)
- [Anular reglas de aprobación \(AWS CLI\)](#)

Anular reglas de aprobación (consola)

Puede reemplazar los requisitos de las reglas de aprobación en una solicitud de extracción en la consola, dentro del proceso de revisión de una solicitud de extracción. Si cambia de opinión, puede revocar la anulación y volver a aplicar los requisitos de la regla de aprobación. Solo se pueden anular reglas de aprobación o revocar una modificación si la solicitud de extracción continúa abierta. Si se ha combinado o cerrado, no se puede cambiar su estado de anulación.

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione el nombre del repositorio.
3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción). Seleccione la solicitud de extracción cuyos requisitos de regla de aprobación desee anular o cuya anulación desee revocar.
4. En la pestaña Approvals (Aprobaciones) elija Override approval rules (Sustituir reglas de aprobación). Los requisitos se dejan de lado y el texto del botón cambia a Revoke override (Revocar anulación). Para volver a aplicar los requisitos de regla de aprobación, seleccione Revoke override (Revocar anulación).

Anular reglas de aprobación (AWS CLI)

Puede utilizar la AWS CLI para anular los requisitos de las reglas de aprobación. También puede usarla para ver el estado de anulación de una solicitud de extracción.

Para sustituir los requisitos de reglas de aprobación en una solicitud de extracción

1. En un terminal o en la línea de comandos, ejecute el comando `override-pull-request-approval-rules` especificando lo siguiente:
 - El ID generado por el sistema de la solicitud de extracción.
 - El ID de revisión más reciente de la solicitud de extracción. Para ver esta información, utilice `get-pull-request`.

- El estado que desea para la anulación: **OVERRIDE** o **REVOKE**. El estado **REVOKE** elimina el estado **OVERRIDE**, pero no se guarda.

Por ejemplo, para anular las reglas de aprobación en una solicitud de extracción con el ID **34** y el ID de revisión **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8EXAMPLE --override-status OVERRIDE
```

2. Si se ejecuta correctamente, este comando no devuelve nada.
3. Para revocar la anulación en una solicitud de extracción con el ID **34** y el ID de revisión **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8EXAMPLE --override-status REVOKE
```

Para obtener información sobre el estado de anulación de una solicitud de extracción

1. En un terminal o en la línea de comandos, ejecute el comando `get-pull-request-override-state` especificando lo siguiente:
 - El ID generado por el sistema de la solicitud de extracción.
 - El ID de revisión más reciente de la solicitud de extracción. Para ver esta información, utilice `get-pull-request`.

Por ejemplo, para ver el estado de anulación de una solicitud de extracción con el ID **34** y el ID de revisión **927df8d8EXAMPLE**:

```
aws codecommit get-pull-request-override-state --pull-request-id 34 --revision-  
id 927df8d8EXAMPLE
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "overridden": true,  
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

Combinación de una solicitud de extracción en un repositorio de AWS CodeCommit

Una vez que se haya revisado el código y se hayan satisfecho todas las reglas de aprobación (si las hay) de la solicitud de extracción, puede combinar una solicitud de extracción de varias maneras:

- Puede usar la consola para combinar automáticamente la ramificación de origen con la de destino utilizando una de las estrategias de fusión disponibles, que también cierra la solicitud de extracción. También puede resolver cualquier conflicto de combinación en la consola. La consola muestra un mensaje que indica si la solicitud de extracción se puede combinar o si se deben resolver conflictos. Cuando se resuelven todos los conflictos y se elige Merge (Combinar), la combinación se realiza utilizando la estrategia de combinación que se elija. El avance rápido es la estrategia de combinación predeterminada, que es la opción predeterminada para Git. Según el estado del código en la ramificación de origen y en la de destino, esta estrategia podría no estar disponible, pero sí otras opciones, como, por ejemplo, combinación de squash o de tres modos.
- Puede usar la AWS CLI para combinar y cerrar la solicitud de extracción mediante la estrategia de combinación de avance rápido, squash o tres modos.
- En el equipo local, puede utilizar el comando `git merge` para combinar la ramificación de origen con la de destino y, a continuación, enviar el código combinado a la ramificación de destino. Este enfoque presenta algunos inconvenientes que es importante considerar cuidadosamente. Combina la solicitud de extracción independientemente de si se han cumplido los requisitos de las normas de aprobación de la solicitud de extracción, eludiendo esos controles. La combinación y la inserción de la ramificación de destino también cierra la solicitud de extracción automáticamente si esta última se combina mediante la estrategia de combinación de avance rápido. Una ventaja de este enfoque es que el comando `git merge` le permite elegir opciones de combinación o estrategias que no están disponibles en la consola de CodeCommit. Para obtener más información acerca de `git merge` y las opciones de combinación, consulte [git-merge](#) o la documentación de Git.

CodeCommit cierra automáticamente una solicitud de extracción si se elimina la ramificación de origen o de destino de la solicitud de extracción.

Temas

- [Combinación de una solicitud de extracción \(consola\)](#)
- [Combinar una solicitud de extracción \(AWS CLI\)](#)

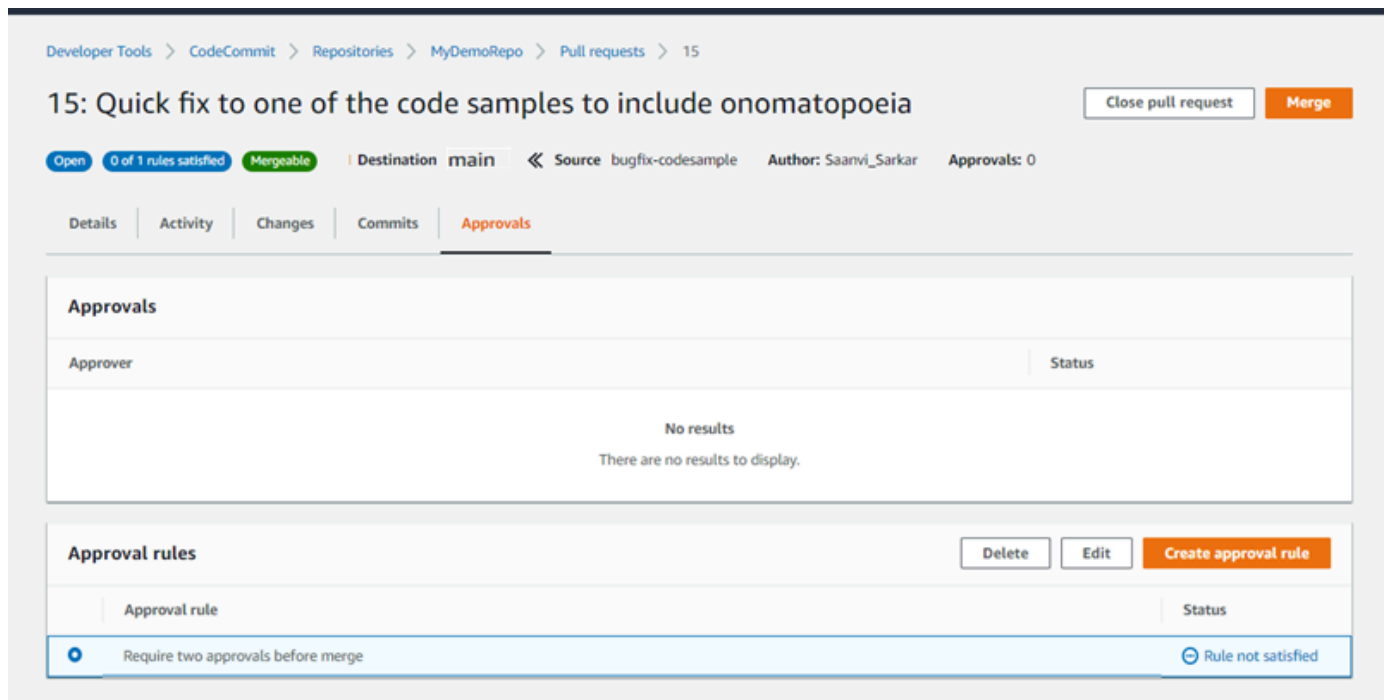
Combinación de una solicitud de extracción (consola)

Puede utilizar la consola de CodeCommit para fusionar una solicitud de extracción en un repositorio de CodeCommit. Después de que el estado de una solicitud de extracción cambie a Merged (Combinado), ya no aparece en la lista de las solicitudes de extracción abiertas. Una solicitud de extracción combinada se clasifica como cerrada. No se puede cambiar de nuevo a Open (Abrir), aunque los usuarios pueden seguir realizando comentarios sobre los cambios y respondiendo a los comentarios. Después de combinar o cerrar una solicitud de extracción, no se puede aprobar, revocar la aprobación para ella ni sustituir las reglas de aprobación aplicadas a la solicitud de extracción.

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione el nombre del repositorio.
3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas. Elija la solicitud de extracción abierta que desea combinar.
5. En la solicitud de extracción, elija Approvals (Aprobaciones). Revise la lista de aprobadores y compruebe que se cumplan las condiciones de todas las reglas de aprobación (si las hay). No se puede combinar una solicitud de extracción si una o varias reglas de aprobación tienen el estado Rule not satisfied (Regla no cumplida). Si nadie ha aprobado la solicitud de extracción, considere si desea combinarla o si desea esperar las aprobaciones.

Note

Si se creó una regla de aprobación para una solicitud de extracción, puede editarla o eliminarla para desbloquear la combinación. Si la regla de aprobación se creó con una plantilla de regla de aprobación, no puede editarla ni eliminarla. Solo podrá optar por anular los requisitos. Para obtener más información, consulte [Anular reglas de aprobación en una solicitud de extracción](#).



6. Seleccione Combinar.
7. En la solicitud de extracción, elija entre las estrategias de combinación disponibles. Las estrategias de combinación que no se pueden aplicar aparecen atenuadas. Si no hay más estrategias de combinación disponibles, puede elegir resolver los conflictos manualmente en la consola de CodeCommit o puede resolverlos localmente mediante su cliente de Git. Para obtener más información, consulte [Resolver conflictos en una solicitud de extracción en un repositorio de AWS CodeCommit](#).

Merge pull request 9: Bug fix for unhandled exception

Merge request details

Pull request: #9 Bug fix for unhandled exception

Destination main **Source** bugfix-bug1234

Merge strategy [Info](#)

Determines the way in which the current pull request will be merged into the destination branch

☐ **Fast forward merge**
`git merge --ff-only`
Merges the branches and moves the destination branch pointer to the tip of the source branch. This is the default merge strategy in Git.



☒ **Squash and merge**
`git merge --squash`
Combine all commits from the source branch into a single merge commit in the destination branch.



☐ **3-way merge**
`git merge --no-ff`
Create a merge commit and adds individual source commits to the destination branch.



Commit message - *optional*

☐ Preview markdown

Squashed commit of the following

commit d49940ad
Author: Li Juan <li_juan@example.com>
Date: Tue May 07 2019 15:12:48 GMT-0700 (Pacific Daylight Time)

Fixing the bug reported in 1234.

Author name

Maria Garcia

Email address

maria_garcia@example.com

☒ Delete source branch bugfix-bug1234 after merging?

Cancel

Merge pull request

- Una combinación de avance rápido avanza la referencia para la ramificación de destino a la confirmación más reciente de la ramificación de origen. Este es el comportamiento predeterminado de Git si es posible. No se crea ninguna confirmación de combinación, pero todo el historial de confirmaciones de la ramificación de origen se conserva como si hubiera ocurrido en la ramificación de destino. Las combinaciones de avance rápido no aparecen como una combinación de ramificaciones en la vista del visualizador de confirmaciones del historial de la ramificación de destino, porque no se crea ninguna confirmación de

combinación. El extremo de la ramificación de origen se reenvía rápidamente al extremo de la ramificación de destino.

- Una combinación de squash crea una confirmación que contiene los cambios en la ramificación de origen y aplica dicha confirmación de squash única a la ramificación de destino. De forma predeterminada, el mensaje de confirmación de dicha confirmación de squash contiene todos los mensajes de confirmación de los cambios en la ramificación de origen. No se conserva ningún historial de confirmaciones individuales de los cambios de la ramificación. Esto puede ayudar a simplificar el historial del repositorio conservando una representación gráfica de la combinación en la vista del visualizador de confirmación del historial de la ramificación de destino.
 - Una combinación de tres modos crea una confirmación de combinación para la combinación en la ramificación de destino, pero también conserva las confirmaciones individuales realizadas en la ramificación de origen como parte del historial de la ramificación de destino. Esto puede ayudar a mantener un historial completo de los cambios en su repositorio.
8. Si elige la estrategia de combinación de squash o tres modos, revise el mensaje de confirmación generado automáticamente y modifíquelo si desea cambiar la información. Añada el nombre y la dirección de correo electrónico para el historial de confirmaciones.
 9. (Opcional) Anule la selección de la opción para eliminar la ramificación de origen como parte de la combinación. De forma predeterminada, se elimina la ramificación de origen cuando se combina una solicitud de extracción.
 10. Elija Merge pull request (Combinar solicitud de extracción) para completar la combinación.

Combinar una solicitud de extracción (AWS CLI)

Para utilizar los comandos de AWS CLI en CodeCommit, instale la AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar la AWS CLI para fusionar solicitudes de extracción en un repositorio de CodeCommit

1. Para evaluar si una solicitud de extracción ha cumplido todas sus reglas de aprobación y está lista para combinarla, ejecute el comando `evaluate-pull-request-approval-rules` especificando lo siguiente:
 - El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El ID de revisión de la solicitud de extracción (mediante `--revision-id` option). Puede obtener el ID de revisión actual de una solicitud de extracción mediante el comando [get-pull-request](#).

Por ejemplo, para evaluar el estado de las reglas de aprobación en una solicitud de extracción con el ID **27** y el ID de revisión **9f29d167EXAMPLE**:

```
aws codecommit evaluate-pull-request-approval-rules --pull-request-id 27 --  
revision-id 9f29d167EXAMPLE
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

 Note

Este resultado indica que una solicitud de extracción no se puede combinar porque no se han cumplido los requisitos de una regla de aprobación. Para combinar esta solicitud de extracción, puede hacer que los revisores la aprueben para que cumpla las condiciones de la regla. Dependiendo de sus permisos y de cómo se creó la regla, es posible que también pueda editarla, anularla o eliminarla. Para obtener más información, consulte [Revisión de una solicitud de extracción](#), [Anular reglas de aprobación en una solicitud de extracción](#) y [Editar o eliminar una regla de aprobación para una solicitud de extracción](#).

2. Para combinar y cerrar una solicitud de extracción utilizando la estrategia de combinación de avance rápido, ejecute el comando `merge-pull-request-by-fast-forward`, especificando:
 - El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El ID completo de la confirmación en el extremo de la ramificación de origen (con la opción `--source-commit-id`).
 - El nombre del repositorio (con la opción `--repository-name`).

Por ejemplo, para combinar y cerrar una solicitud de extracción con el ID **47** y el ID de confirmación de origen **99132ab0EXAMPLE** en un repositorio llamado **MyDemoRepo**:

```
aws codecommit merge-pull-request-by-fast-forward --pull-request-id 47 --source-commit-id 99132ab0EXAMPLE --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "I want one approver for this pull request",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.142,
    "description": "Review the latest changes and updates to the global variables",
    "lastActivityDate": 1508887223.155,
    "pullRequestId": "47",
    "pullRequestStatus": "CLOSED",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": true,
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
```

```

        "sourceReference": "refs/heads/variables-branch"
      }
    ],
    "title": "Consolidation of global variables"
  }
}

```

3. Para combinar y cerrar una solicitud de extracción mediante la estrategia de combinación de squash, ejecute el comando `merge-pull-request-by-squash`, especificando:
 - El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El ID completo de la confirmación en el extremo de la ramificación de origen (con la opción `--source-commit-id`).
 - El nombre del repositorio (con la opción `--repository-name`).
 - El nivel de detalle de conflictos que desea utilizar (con la opción `--conflict-detail-level`). Si no se especifica, se utiliza el **FILE_LEVEL** predeterminado.
 - La estrategia de resolución de conflictos que desea utilizar (con la opción `--conflict-resolution-strategy`). Si no se especifica, toma de forma predeterminada el valor **NONE** y los conflictos deben resolverse manualmente.
 - El mensaje de confirmación que incluir (con la opción `--commit-message`).
 - El nombre que se utilizará para la confirmación (con la opción `--author-name`).
 - La dirección de correo electrónico que se va a utilizar para la confirmación (con la opción `--email`).
 - Si desea mantener cualquier carpeta vacía (con la opción `--keep-empty-folders`).

En el ejemplo siguiente se combina y cierra una solicitud de extracción con el ID **47** y el ID de confirmación de origen **99132ab0EXAMPLE** en un repositorio denominado **MyDemoRepo**. Utiliza los detalles de conflictos **LINE_LEVEL** y la estrategia de resolución de conflictos de **ACCEPT_SOURCE**:

```

aws codecommit merge-pull-request-by-squash --pull-request-id 47 --source-commit-id 99132ab0EXAMPLE --repository-name MyDemoRepo --conflict-detail-level LINE_LEVEL --conflict-resolution-strategy ACCEPT_SOURCE --author-name "Jorge Souza" --email "jorge_souza@example.com" --commit-message "Merging pull request 47 by squash and accepting source in merge conflicts"

```

Si se ejecuta correctamente, este comando produce el mismo tipo de salida que la combinación de avance rápido, un resultado similar al siguiente:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
        \"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type
        \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
        [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 1571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
          "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.142,
    "description": "Review the latest changes and updates to the global
    variables",
    "lastActivityDate": 1508887223.155,
    "pullRequestId": "47",
    "pullRequestStatus": "CLOSED",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": true,
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
      }
    ]
  }
}
```

```
        }
      ],
      "title": "Consolidation of global variables"
    }
  }
}
```

4. Para combinar y cerrar una solicitud de extracción mediante estrategia de combinación de tres modos, ejecute el comando `merge-pull-request-by-three-way`, especificando:
- El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El ID completo de la confirmación en el extremo de la ramificación de origen (con la opción `--source-commit-id`).
 - El nombre del repositorio (con la opción `--repository-name`).
 - El nivel de detalle de conflictos que desea utilizar (con la opción `--conflict-detail-level`). Si no se especifica, se utiliza el **FILE_LEVEL** predeterminado.
 - La estrategia de resolución de conflictos que desea utilizar (con la opción `--conflict-resolution-strategy`). Si no se especifica, toma de forma predeterminada el valor **NONE** y los conflictos deben resolverse manualmente.
 - El mensaje de confirmación que incluir (con la opción `--commit-message`).
 - El nombre que se utilizará para la confirmación (con la opción `--author-name`).
 - La dirección de correo electrónico que se va a utilizar para la confirmación (con la opción `--email`).
 - Si desea mantener cualquier carpeta vacía (con la opción `--keep-empty-folders`).

En el ejemplo siguiente se combina y cierra una solicitud de extracción con el ID **47** y el ID de confirmación de origen **99132ab0EXAMPLE** en un repositorio denominado **MyDemoRepo**. Utiliza las opciones predeterminadas para los detalles de conflictos y la estrategia de resolución de conflictos:

```
aws codecommit merge-pull-request-by-three-way --pull-request-id 47 --source-commit-id 99132ab0EXAMPLE --repository-name MyDemoRepo --author-name "Maria Garcia" --email "maria_garcia@example.com" --commit-message "Merging pull request 47 by three-way with default options"
```

Si se ejecuta correctamente, este comando produce el mismo tipo de salida que la combinación de avance rápido, de forma similar a como se muestra a continuación:

```

{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\\\"DestinationReferences\\\": [\\\"refs/heads/main\\\"],\\\"Statements\\\": [{\\\"Type
\\\": \\\"Approvers\\\",\\\"NumberOfApprovalsNeeded\\\": 2,\\\"ApprovalPoolMembers\\\":
[\\\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\\\"]}]}\",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 1571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
          "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.142,
    "description": "Review the latest changes and updates to the global
variables",
    "lastActivityDate": 1508887223.155,
    "pullRequestId": "47",
    "pullRequestStatus": "CLOSED",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": true,
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
      }
    ],
    "title": "Consolidation of global variables"
  }
}

```



```
}
```

Resolver conflictos en una solicitud de extracción en un repositorio de AWS CodeCommit

Si su solicitud de extracción tiene conflictos y no se puede combinar, puede intentar resolver los conflictos de varias formas:

- En el equipo local, puede utilizar el comando `git diff` para encontrar los conflictos entre las dos ramificaciones y realizar cambios para resolverlos. También puede utilizar una herramienta de diferencias u otro software que le ayudarán a encontrar y resolver las diferencias. Una vez que las haya resuelto de forma satisfactoria, puede enviar su ramificación de origen con los cambios que contienen los conflictos resueltos, lo que actualizará la solicitud de extracción. Para obtener más información sobre `git diff` y `git difftool`, consulte la documentación de Git.
- En la consola, puede elegir **Resolve conflicts (Resolver conflictos)**. Esto abre un editor de texto sin formato que muestra los conflictos de forma similar al comando `git diff`. Puede revisar manualmente los conflictos de cada archivo que los contenga, realizar cambios y, a continuación, actualizar la solicitud de extracción con los cambios.
- En el AWS CLI, puede utilizar la AWS CLI para obtener información sobre los conflictos de combinación y crear una confirmación de combinación sin referencias para probar una combinación.

Temas

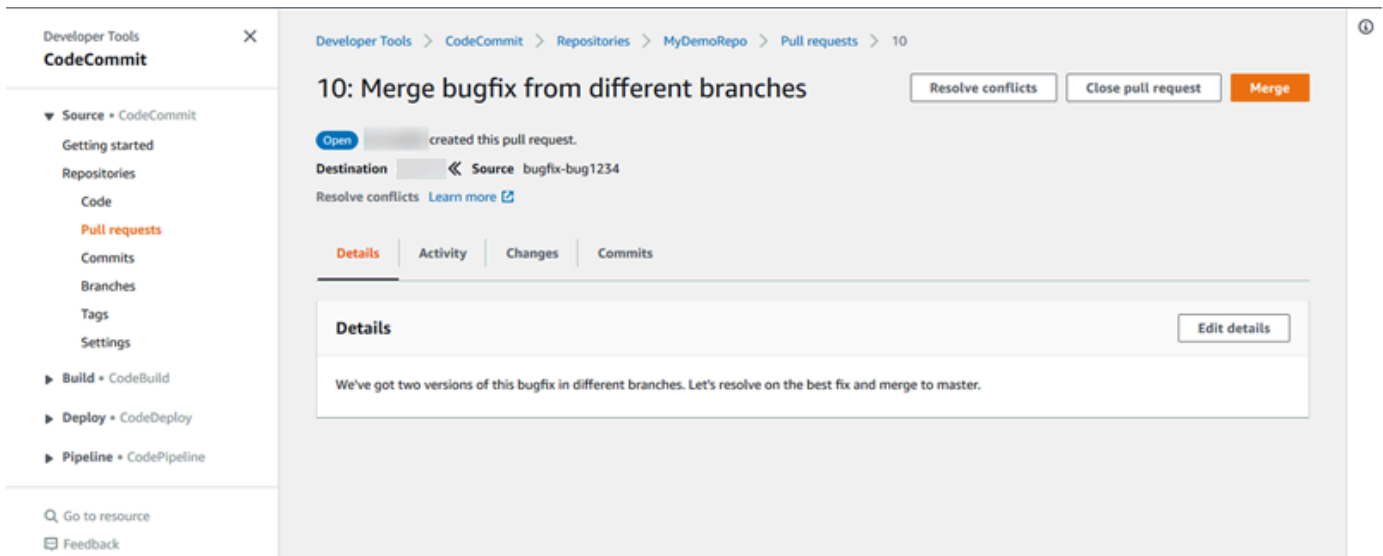
- [Resolución de conflictos en una solicitud de extracción \(consola\)](#)
- [Resolución de conflictos en una solicitud de extracción \(AWS CLI\)](#)

Resolución de conflictos en una solicitud de extracción (consola)

Puede utilizar la consola de CodeCommit para resolver conflictos en una solicitud de extracción en un repositorio de CodeCommit.

1. Abra la consola de CodeCommit en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En **Repositories (Repositorios)**, seleccione el nombre del repositorio.

3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas. Elija la solicitud de extracción abierta que desea combinar pero que contiene conflictos.
5. En la solicitud de extracción, elija Resolve conflicts (Resolver conflictos). Esta opción solo aparece si hay conflictos que deben resolverse antes de que la solicitud de extracción se pueda combinar.



6. Se abre una ventana de resolución de conflictos que muestra todos los archivos que tienen conflictos que deban resolverse. Elija cada archivo de la lista para revisar los conflictos y haga los cambios necesarios hasta que se hayan resuelto todos los conflictos.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 10 > Resolve conflicts

Resolve conflicts

Resolve conflicts in each of the files in the list. When you have resolved all conflicts, update the pull request and review the merge strategies available. [Info](#)

Destination << Source bugfix-bug1234

Editing: helloworld.py

Reset file

Delete file

Use source content

Use destination content

helloworld.py

1

```

1 import sys
2
3 print('Hello, World!')
4
5 <<<<<< HEAD:helloworld.py
6 print('The sum of 2 and 3 is 5.')
7 =====
8 print('The sum of 3 and 2 is 5.')
9 >>>>>> bugfix-bug1234:helloworld.py
10
11 sum = int(sys.argv[1]) + int(sys.argv[2])
12
13 print('The sum of {0} and {1} is {2}.'.format(sys.argv[1], sys.argv[2], sum))

```

☐ Allow the merge to proceed with Git conflict markers still present.

Cancel

Update pull request

- Puede optar por usar el contenido del archivo de origen, el contenido del archivo de destino o si el archivo no es un archivo binario, para editar manualmente el contenido de un archivo para que contenga únicamente los cambios que desee. Se utilizan marcadores de diferencia estándar de git para mostrar los marcadores de conflictos entre las ramificaciones de destino (HEAD) y de origen en el archivo.
- Si un archivo es un archivo binario, un submódulo de Git o si hay un conflicto de nombre de archivo/carpeta, debe optar por utilizar el archivo de origen o el archivo de destino para resolver los conflictos. No puede ver o editar archivos binarios en la consola de CodeCommit.
- Si hay conflictos de modo de archivo, verá la opción para resolver dicho conflicto eligiendo entre el modo de archivo del archivo de origen y el modo de archivo del archivo de destino.

- Si decide que desea descartar los cambios para un archivo y restaurarlos a su estado en conflicto, elija Reset file (Restablecer archivo). Esto le permite resolver los conflictos en una manera diferente.
7. Cuando esté satisfecho con los cambios, elija Undate pull request (Actualizar solicitud de extracción).

 Note

Debe resolver todos los conflictos de todos los archivos para que pueda actualizar correctamente la solicitud de extracción con los cambios.

8. La solicitud de extracción se actualiza con los cambios y se puede combinar. Verá la página de combinación. Puede elegir combinar la solicitud de extracción en este momento o puede volver a la lista de solicitudes de extracción.

Resolución de conflictos en una solicitud de extracción (AWS CLI)

Para utilizar los comandos de AWS CLI en CodeCommit, instale la AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Ningún comando de AWS CLI único le permitirá resolver los conflictos en una solicitud de extracción y combinar dicha solicitud. Sin embargo, puede utilizar los comandos individuales para descubrir los conflictos, intentar resolverlos y comprobar si una solicitud de extracción es combinable. Puede usar:

- `get-merge-options`, para averiguar qué opciones de combinación están disponibles para combinar entre dos especificadores de confirmación.
- `get-merge-conflicts`, para devolver una lista de archivos con conflictos de combinación en una combinación entre dos especificadores de confirmación.
- `batch-describe-merge-conflicts`, para obtener información acerca de todos los conflictos de combinación en los archivos en una fusión entre dos confirmaciones mediante una estrategia de fusión especificada.
- `describe-merge-conflicts`, para obtener información detallada sobre conflictos de combinación para un archivo específico entre dos confirmaciones mediante una estrategia de combinación especificada.
- `create-unreferenced-merge-commit`, para probar el resultado de combinar dos especificadores de combinación mediante una estrategia de combinación especificada.

1.

Para descubrir qué opciones de fusión están disponibles para una combinación de dos especificadores de confirmación, ejecute el comando `get-merge-options`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- El nombre del repositorio (con la opción `--repository-name`).
- (Opcional) Una estrategia de resolución de conflictos que se va a utilizar (con la opción `--conflict-resolution-strategy`).
- (Opcional) El nivel de detalle que desea sobre los conflictos (con la opción `--conflict-detail-level`).

Por ejemplo, para determinar las estrategias de combinación disponibles para combinar una ramificación de origen denominada *bugfix-1234* con una ramificación de destino denominada *principal* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit get-merge-options --source-commit-specifier bugfix-1234 --  
destination-commit-specifier main --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "mergeOptions": [  
    "FAST_FORWARD_MERGE",  
    "SQUASH_MERGE",  
    "THREE_WAY_MERGE"  
  ],  
  "sourceCommitId": "d49940adEXAMPLE",  
  "destinationCommitId": "86958e0aEXAMPLE",  
  "baseCommitId": "86958e0aEXAMPLE"  
}
```

2.

Para obtener una lista de los archivos que contienen conflictos de combinación en una combinación entre dos especificadores de confirmación, ejecute el comando `get-merge-conflicts`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- El nombre del repositorio (con la opción `--repository-name`).
- La opción de combinación que desea utilizar (con la opción `--merge-option`).
- (Opcional) El nivel de detalle que desea sobre los conflictos (con la opción `--conflict-detail-level`).
- (Opcional) Una estrategia de resolución de conflictos que se va a utilizar (con la opción `--conflict-resolution-strategy`).
- (Opcional) El número máximo de archivos con conflictos que devolver (con la opción `--max-conflict-files`).

Por ejemplo, para obtener una lista de archivos que contienen conflictos en una combinación entre una ramificación de origen denominada `feature-randomizationfeature` y una ramificación de destino denominada `principal` con la estrategia de combinación de tres modos en un repositorio denominado `MyDemoRepo`:

```
aws codecommit get-merge-conflicts --source-commit-specifier feature-
randomizationfeature --destination-commit-specifier main --merge-option
THREE_WAY_MERGE --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "mergeable": false,
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b6958EXAMPLE",
  "conflictMetadataList": [
    {
      "filePath": "readme.md",
      "fileSizes": {
        "source": 139,
        "destination": 230,
        "base": 85
      }
    }
  ]
}
```

```

        "fileModes": {
            "source": "NORMAL",
            "destination": "NORMAL",
            "base": "NORMAL"
        },
        "objectTypes": {
            "source": "FILE",
            "destination": "FILE",
            "base": "FILE"
        },
        "numberOfConflicts": 1,
        "isBinaryFile": {
            "source": false,
            "destination": false,
            "base": false
        },
        "contentConflict": true,
        "fileModeConflict": false,
        "objectTypeConflict": false,
        "mergeOperations": {
            "source": "M",
            "destination": "M"
        }
    }
}

```

3.

Para obtener información sobre los conflictos de combinación en todos los archivos o un subconjunto de archivos en una combinación de dos especificadores de confirmación, ejecute el comando `batch-describe-merge-conflicts`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- La opción de combinación que desea utilizar (con la opción `--merge-option`).
- El nombre del repositorio (con la opción `--repository-name`).
- (Opcional) Una estrategia de resolución de conflictos que se va a utilizar (con la opción `--conflict-resolution-strategy`).

- (Opcional) El nivel de detalle que desea sobre los conflictos (con la opción `--conflict-detail-level`).
- (Opcional) El número máximo de fragmentos de combinación que devolver (con la opción `--max-merge-hunks`).
- (Opcional) El número máximo de archivos con conflictos que devolver (con la opción `--max-conflict-files`).
- (Opcional) La ruta de archivos de destino que se va a utilizar para describir los conflictos (con la opción `--file-paths`).

Por ejemplo, para determinar los conflictos de combinación para combinar una ramificación de origen denominada *feature-randomizationfeature* con una ramificación de destino denominada *principal* utilizando la estrategia *COMBINACIÓN_TRES_MODOS* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit batch-describe-merge-conflicts --source-commit-specifier feature-randomizationfeature --destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "conflicts": [
    {
      "conflictMetadata": {
        "filePath": "readme.md",
        "fileSizes": {
          "source": 139,
          "destination": 230,
          "base": 85
        },
        "fileModes": {
          "source": "NORMAL",
          "destination": "NORMAL",
          "base": "NORMAL"
        },
        "objectTypes": {
          "source": "FILE",
          "destination": "FILE",
          "base": "FILE"
        }
      }
    }
  ]
}
```



```

    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
        "source": false,
        "destination": false,
        "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
        "source": "M",
        "destination": "M"
    }
},
"mergeHunks": [
    {
        "isConflict": true,
        "source": {
            "startLine": 0,
            "endLine": 3,
            "hunkContent": "VGhpcyBpEXAMPLE=="
        },
        "destination": {
            "startLine": 0,
            "endLine": 1,
            "hunkContent": "VXNlIHRoEXAMPLE="
        }
    }
]
},
"errors": [],
"destinationCommitId": "86958e0aEXAMPLE",
"sourceCommitId": "6ccd57fdEXAMPLE",
"baseCommitId": "767b6958EXAMPLE"
}

```

4.

Para obtener información detallada sobre conflictos de combinación para un archivo específico en una combinación de dos especificadores de confirmación, ejecute el comando `describe-merge-conflicts`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- La opción de combinación que desea utilizar (con la opción `--merge-option`).
- La ruta del archivo de destino que se va a utilizar para describir los conflictos (con la opción `--file-path`).
- El nombre del repositorio (con la opción `--repository-name`).
- (Opcional) Una estrategia de resolución de conflictos que se va a utilizar (con la opción `--conflict-resolution-strategy`).
- (Opcional) El nivel de detalle que desea sobre los conflictos (con la opción `--conflict-detail-level`).
- (Opcional) El número máximo de fragmentos de combinación que devolver (con la opción `--max-merge-hunks`).
- (Opcional) El número máximo de archivos con conflictos que devolver (con la opción `--max-conflict-files`).

Por ejemplo, para determinar los conflictos de combinación para un archivo denominado *readme.md* en una ramificación de origen denominada *feature-randomizationfeature* con una ramificación de destino denominada *principal* utilizando la estrategia *COMBINACIÓN_TRES_MODOS* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit describe-merge-conflicts --source-commit-specifier feature-randomizationfeature --destination-commit-specifier main --merge-option THREE_WAY_MERGE --file-path readme.md --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "conflictMetadata": {
    "filePath": "readme.md",
    "fileSizes": {
      "source": 139,
      "destination": 230,
      "base": 85
    }
  },
}
```

```
    "fileModes": {
      "source": "NORMAL",
      "destination": "NORMAL",
      "base": "NORMAL"
    },
    "objectTypes": {
      "source": "FILE",
      "destination": "FILE",
      "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
      "source": false,
      "destination": false,
      "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  },
  "mergeHunks": [
    {
      "isConflict": true,
      "source": {
        "startLine": 0,
        "endLine": 3,
        "hunkContent": "VGhpcyBpEXAMPLE=="
      },
      "destination": {
        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ],
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b69580EXAMPLE"
}
```

5.

Para crear una confirmación sin referencia que representa el resultado de combinar dos especificadores, ejecute el comando `create-unreferenced-merge-commit`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- La opción de combinación que desea utilizar (con la opción `--merge-option`).
- El nombre del repositorio (con la opción `--repository-name`).
- (Opcional) Una estrategia de resolución de conflictos que se va a utilizar (con la opción `--conflict-resolution-strategy`).
- (Opcional) El nivel de detalle que desea sobre los conflictos (con la opción `--conflict-detail-level`).
- (Opcional) El mensaje de confirmación que incluir (con la opción `--commit-message`).
- (Opcional) El nombre que se va a utilizar para la confirmación (con la opción `--name`).
- (Opcional) La dirección de correo electrónico para la confirmación (con la opción `--email`).
- (Opcional) Si cualquier vacío para mantener carpetas (con la opción `--keep-empty-folders`).

Por ejemplo, para determinar los conflictos de combinación para combinar una ramificación de origen denominada *bugfix-1234* con una ramificación de destino denominada principal utilizando la estrategia *ACEPTAR_ORIGEN* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit create-unreferenced-merge-commit --source-commit-specifier bugfix-1234 --destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-name MyDemoRepo --name "Maria Garcia" --email "maria_garcia@example.com" --commit-message "Testing the results of this merge."
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Cerrar una solicitud de extracción en un AWS CodeCommit repositorio

Si desea cerrar una solicitud de extracción sin combinar el código, puede hacerlo de varias formas:

- En la consola, puede cerrar una solicitud de extracción sin combinar el código. Se recomienda hacerlo si desea utilizar el comando `git merge` para combinar manualmente las ramificaciones o si no desea combinar el código de la ramificación de origen de la solicitud de extracción en la ramificación de destino.
- Puedes eliminar la rama de origen especificada en la solicitud de extracción. CodeCommit cierra una solicitud de extracción automáticamente si se elimina la rama de origen o destino de la solicitud de extracción.
- En el AWS CLI, puedes actualizar el estado de una solicitud de extracción de `OPEN` a `CLOSED`. Esto cierra la solicitud de extracción sin combinar el código.

Temas

- [Cierre de una solicitud de extracción \(consola\)](#)
- [Cerrar una solicitud de extracción \(AWS CLI\)](#)

Cierre de una solicitud de extracción (consola)

Puedes usar la CodeCommit consola para cerrar una solicitud de extracción en un CodeCommit repositorio. Después de que el estado de una solicitud de extracción cambie a `Closed`, ya no se puede cambiar de nuevo a `Open`, aunque los usuarios pueden seguir realizando comentarios sobre los cambios y respondiendo a los comentarios.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), seleccione el nombre del repositorio.
3. En el panel de navegación, seleccione Pull requests (Solicitudes de extracción).
4. De forma predeterminada, se muestra una lista de todas las solicitudes de extracción abiertas. Elija la solicitud de extracción abierta que desea cerrar.

Pull requests

Info

Open pull requests

Create pull request

<

1

>

Pull request	Author	Destination	Last activity	Status	Approval status
31: testing this	Saanvi_Sarkar	preprod	4 minutes ago	Open	No approval rules
25: Updated some of our Java samples	Li_Juan	main	5 minutes ago	Open	0 of 1 rules satisfied
29: Changing duplicate value	Li_Juan	main	19 minutes ago	Open	0 of 1 rules satisfied
22: Test pull request	Saanvi_Sarkar	preprod	20 minutes ago	Open	No approval rules
28: Changes to some of our code samples	Li_Juan	main	1 month ago	Open	0 of 1 rules satisfied
20: A bugfix to add missing examples for S3	Saanvi_Sarkar	main	1 month ago	Open	0 of 1 rules satisfied

- En la solicitud de extracción, elija Close pull request (Cerrar solicitud de extracción). Esta opción cierra la solicitud de extracción sin intentar combinar la ramificación de origen en la de destino. Aunque esta opción no ofrece la posibilidad de eliminar la ramificación de origen como parte del proceso de cierre de la solicitud de extracción, puede hacerlo usted mismo después de que se cierre la solicitud.

Cerrar una solicitud de extracción (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar el AWS CLI para cerrar las solicitudes de cambios en un CodeCommit repositorio

- Para actualizar el estado de una solicitud de extracción en un repositorio de OPEN a CLOSED, ejecute el comando `update-pull-request-status` especificando lo siguiente:
 - El ID de la solicitud de extracción (con la opción `--pull-request-id`).
 - El estado de la solicitud de extracción (con la opción `--pull-request-status`).

Por ejemplo, para actualizar el estado de una solicitud de extracción con el ID de **42** a un estado de **CLOSED** en un CodeCommit repositorio llamado `MyDemoRepo`:

```
aws codecommit update-pull-request-status --pull-request-id 42 --pull-request-status CLOSED
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codecommit:GetRepository",
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ]
    }
  ]
}
```

Trabajar con plantillas de reglas de aprobación

Puede crear reglas de aprobación para solicitudes de extracción. Para aplicar automáticamente reglas de aprobación a algunas o todas las solicitudes de extracción creadas en los repositorios, utilice plantillas de reglas de aprobación. Las plantillas de reglas de aprobación le ayudan a personalizar los flujos de trabajo de desarrollo en los repositorios para que las distintas ramificaciones tengan niveles adecuados de aprobación y control. Puede definir diferentes reglas para las ramificaciones de producción y desarrollo. Estas reglas se aplican cada vez que se crea una solicitud de extracción que coincide con las condiciones de la regla. Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Puede asociar una plantilla de reglas de aprobación a uno o más repositorios en el Región de AWS lugar donde se crearon. Cuando una plantilla está asociada con un repositorio, crea automáticamente reglas de aprobación para las solicitudes de extracción en ese repositorio durante la creación de la solicitud de extracción. Al igual que una regla de aprobación única, una plantilla de regla de aprobación define una estructura de regla de aprobación, que incluye el número de aprobaciones necesarias y un grupo opcional de usuarios de los que deben proceder las aprobaciones. A diferencia de una regla de aprobación, también puede definir referencias de destino (la ramificación o las ramificaciones), que también se denominan filtros de ramificación. Si define referencias de destino, solo se crearán reglas para aquellas solicitudes de extracción cuyos nombres de ramificaciones de destino coincidan con los nombres de ramificación especificados (referencias de destino) en la plantilla. Por ejemplo, si especifica **refs/heads/main** como referencia de destino, la regla de aprobación definida en la plantilla solo se aplica a las solicitudes de extracción si la ramificación de destino es `main`.

Approval rule template

Approval rule template name

Require 1 approver from a senior developer for main branch

Description - *optional*

Before a pull request can be merged to the main branch, at least one senior developer must give an approval to the changes.

Number of approvals needed

1

Approval pool members - *optional*

If approval pool members are specified, only approvals from these members will count toward satisfying this rule. You can use wildcards to match multiple approvers with one value.

Approver type [Info](#)

Value

IAM user name or assumed role ▼

SeniorDevelopers/*

Remove

Fully qualified ARN ▼

:s:iam::123456789012:user/Mary_Major

Remove

Add

Branch filters - *optional*

Use branch filters to only apply this template to a pull request if the destination branch name matches a name in the filter list.

Branch name

main

Remove

Add

▼ Associated repositories

Repositories - *optional*

MyDemoRepo ✕

MyTestRepo ✕

Temas

- [Crear una plantilla de regla de aprobación](#)
- [Asociar una plantilla de regla de aprobación con un repositorio](#)
- [Administrar plantillas de reglas de aprobación](#)
- [Desasociar una plantilla de regla de aprobación](#)
- [Eliminar una plantilla de regla de aprobación](#)

Crear una plantilla de regla de aprobación

Puede crear una o varias plantillas de reglas de aprobación para ayudarle a personalizar los flujos de trabajo de desarrollo en los repositorios. Crear varias plantillas le permite configurar la creación automática de reglas de aprobación para que las distintas ramificaciones tengan los niveles adecuados de aprobación y control. Por ejemplo, puede crear diferentes plantillas para las ramificaciones de producción y desarrollo y aplicarlas a uno o más repositorios. Cuando los usuarios crean solicitudes de extracción en esos repositorios, la solicitud se evalúa en relación con esas plantillas. Si la solicitud coincide con las condiciones de las plantillas aplicadas, se crean reglas de aprobación para la solicitud de extracción.

Puede utilizar la consola o AWS CLI crear plantillas de reglas de aprobación. Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Temas

- [Crear una plantilla de regla de aprobación \(consola\)](#)
- [Crear una plantilla de regla de aprobación \(AWS CLI\)](#)

Crear una plantilla de regla de aprobación (consola)

Las plantillas de reglas de aprobación no están asociadas con ningún repositorio de forma predeterminada. Puede establecer una asociación entre una plantilla y uno o varios repositorios al crear la plantilla, o bien puede agregar las asociaciones más adelante.

Para crear una plantilla de regla de aprobación (consola)

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. Elija Approval rule templates (Plantillas de reglas de aprobación) y, a continuación, elija Create template (Crear plantilla).
3. En Approval rule template name (Nombre de plantilla de regla de aprobación), asigne a la plantilla un nombre descriptivo con el fin de saber para qué sirve. Por ejemplo, si desea requerir que una persona de un conjunto de desarrolladores sénior apruebe una solicitud de extracción para que se pueda combinar, puede asignar a la regla el nombre **Require 1 approver from a senior developer**.
4. (Opcional) En Description (Descripción), proporcione una descripción del propósito de esta plantilla. Esto puede ayudar a otros a decidir si esta plantilla es adecuada para sus repositorios.
5. En Number of approvals needed (Número de aprobaciones necesarias), introduzca el número que desee. El valor predeterminado es 1.
6. (Opcional) Si desea exigir que las aprobaciones de una solicitud de extracción procedan de un grupo específico de usuarios, en Approval rule members (Miembros de regla de aprobación), elija Add (Añadir). En Approver type (Tipo de aprobador), elija una de las opciones siguientes:
 - Nombre de usuario de IAM o rol asumido: esta opción rellena previamente el ID de cuenta de Amazon Web Services de la cuenta que utilizó para iniciar sesión y solo requiere un nombre. Se puede utilizar tanto para usuarios de IAM como para usuarios de acceso federado cuyo nombre coincida con el nombre proporcionado. Se trata de una opción muy potente que ofrece una gran flexibilidad. Por ejemplo, si elige esta opción, ha iniciado sesión con la cuenta de Amazon Web Services 123456789012 y especifica **Mary_Major**, todas las opciones siguientes se contabilizarán como aprobaciones procedentes de ese usuario:
 - Un usuario de IAM en la cuenta (arn:aws:iam::123456789012:user/Mary_Major)
 - Un usuario federado identificado en IAM como Mary_Major (arn:aws:sts::123456789012:federated-user/Mary_Major)

Esta opción no reconoce una sesión activa de alguien que asuma el rol **CodeCommitReview** con el nombre de sesión de rol Mary_Major (arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major) a menos que incluya un comodín (*Mary_Major). También puede especificar el nombre del rol explícitamente (CodeCommitReview/Mary_Major).

 - ARN completo: esta opción le permite especificar el nombre de recurso de Amazon (ARN) completo del usuario de IAM o rol. Esta opción también admite las funciones asumidas que utilizan otros AWS servicios, como y. AWS Lambda AWS CodeBuild Para los roles asumidos, el formato de ARN debe ser arn:aws:sts::*AccountID*:assumed-role/*RoleName*

para los roles y `arn:aws:sts::AccountID:assumed-role/FunctionName` para las funciones.

Si eligió el nombre de usuario de IAM o el rol asumido como tipo de aprobador, en Valor, introduzca el nombre del usuario de IAM o rol o el ARN completo del usuario o rol. Elija Add (Agregar) de nuevo para agregar más usuarios o roles, hasta que haya agregado todos los usuarios o roles cuyas aprobaciones se contabilizan en el número de aprobaciones necesarias.

Ambos tipos de aprobadores le permiten utilizar comodines (*) en los valores. Por ejemplo, si elige la opción de nombre de usuario de IAM o rol asumido y especifica **CodeCommitReview/***, todos los usuarios que asumen el rol de **CodeCommitReview** se cuentan en el grupo de aprobación. Sus nombres de sesión de rol individuales se contabilizan en el número necesario de aprobadores. De esta manera, tanto Mary_Major como Li_Juan se contabilizan como aprobaciones al iniciar sesión y asumir el rol de CodeCommitReview. Para obtener más información sobre IAM ARNs, los caracteres comodín y los formatos, consulte [Identificadores de IAM](#).

 Note

Las reglas de aprobación no admiten aprobaciones entre cuentas.

7. (Opcional) En Branch filters (Filtros de ramificación), introduzca los nombres de ramificación de destino para filtrar la creación de reglas de aprobación. Por ejemplo, si lo especifica *main*, solo se creará una regla de aprobación para las solicitudes de extracción en los repositorios asociados si la rama de destino de la solicitud de extracción es una rama denominada. *main*. Puede utilizar comodines (*) en los nombres de ramificación para aplicar reglas de aprobación a todos los nombres de ramificación que coincidan con los casos de comodín. Sin embargo, no se puede utilizar un comodín al principio del nombre de una ramificación. Puede especificar hasta 100 nombres de ramificación. Si no especifica ningún filtro, la plantilla se aplicará a todas las ramificaciones de un repositorio asociado.
8. (Opcional) En Repositorios asociados, en la lista Repositorios elija los repositorios de Región de AWS que desea asociar a esta regla de aprobación.

 Note

Puede optar por asociar repositorios después de crear la plantilla. Para obtener más información, consulte [Asociar una plantilla de regla de aprobación con un repositorio](#).

9. Elija Crear.

Approval rule template

Approval rule template name

Require 1 approver from a senior developer for main branch

Description - *optional*

Before a pull request can be merged to the main branch, at least one senior developer must give an approval to the changes.

Number of approvals needed

1

Approval pool members - *optional*

If approval pool members are specified, only approvals from these members will count toward satisfying this rule. You can use wildcards to match multiple approvers with one value.

Approver type [Info](#)

Value

IAM user name or assumed role ▼

SeniorDevelopers/*

Remove

Fully qualified ARN ▼

:s:iam::123456789012:user/Mary_Major

Remove

Add

Branch filters - *optional*

Use branch filters to only apply this template to a pull request if the destination branch name matches a name in the filter list.

Branch name

main

Remove

Add

▼ Associated repositories

Repositories - *optional*

MyDemoRepo ✕

MyTestRepo ✕

Crear una plantilla de regla de aprobación (AWS CLI)

Puedes utilizarla AWS CLI para crear plantillas de reglas de aprobación. Al usar el AWS CLI, puedes especificar las referencias de destino para la plantilla, de modo que solo se apliquen a las solicitudes de extracción cuyas sucursales de destino coincidan con las de la plantilla.

Para crear una plantilla de regla de aprobación (AWS CLI)

1. En un terminal o en la línea de comandos, ejecute el comando `create-approval-rule-template` especificando lo siguiente:
 - El nombre de la plantilla de regla de aprobación. Es conveniente usar un nombre que describa su propósito.
 - Una descripción de la plantilla de regla de aprobación. Al igual que con el nombre, es conveniente proporcionar una descripción detallada.
 - La estructura JSON de la plantilla de regla de aprobación. Esta estructura puede incluir los requisitos para las referencias de destino, que son las ramificaciones de destino para las solicitudes de extracción en las que se aplica la regla de aprobación, así como miembros del grupo de aprobación, que son usuarios cuyas aprobaciones se contabilizan en el número de aprobaciones necesarias.

Al crear el contenido de la regla de aprobación, puede especificar a los aprobadores de un grupo de aprobación de dos maneras:

- **CodeCommitApprovers**: Esta opción solo requiere una cuenta de Amazon Web Services y un recurso. Se puede utilizar tanto para usuarios de IAM como para usuarios de acceso federado cuyo nombre coincida con el nombre de recurso proporcionado. Se trata de una opción muy potente que ofrece una gran flexibilidad. Por ejemplo, si especifica la cuenta 123456789012 de AWS y **Mary_Major**, todas las opciones siguientes se contabilizan como aprobaciones procedentes de ese usuario:
 - Un usuario de IAM en la cuenta (`arn:aws:iam::123456789012:user/Mary_Major`)
 - Un usuario federado identificado en IAM como **Mary_Major** (`arn:aws:sts::123456789012:federated-user/Mary_Major`)

Esta opción no reconoce una sesión activa de alguien que asuma el rol *SeniorDevelopers* con el nombre de sesión de rol *Mary_Major* (`arn:aws:sts::123456789012:assumed-`

role/*SeniorDevelopers/Mary_Major*) a menos que incluya un comodín (*Mary_Major).

- ARN completo: esta opción le permite especificar el nombre de recurso de Amazon (ARN) completo del usuario de IAM o rol.

[Para obtener más información sobre la IAM ARNs, los caracteres comodín y los formatos, consulte Identificadores de IAM.](#)

En el ejemplo siguiente se crea una plantilla de regla de aprobación denominada **2-approver-rule-for-main** y la descripción **Requires two developers from the team to approve the pull request if the destination branch is main**. La plantilla requiere que dos usuarios que asuman el rol de **CodeCommitReview** aprueben cualquier solicitud de extracción antes de que se pueda combinar con la ramificación **main**:

```
aws codecommit create-approval-rule-template --approval-rule-template-name 2-
approver-rule-for-main --approval-rule-template-description "Requires two
developers from the team to approve the pull request if the destination branch
is main" --approval-rule-template-content "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "2-approver-rule-for-main",
    "creationDate": 1571356106.936,
    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "Requires two developers from the team
to approve the pull request if the destination branch is main",
    "lastModifiedDate": 1571356106.936,
    "ruleContentSha256": "4711b576EXAMPLE"
  }
}
```


Asociar una plantilla de regla de aprobación con un repositorio

Las plantillas de reglas de aprobación se crean en un repositorio específico Región de AWS, pero no afectan a ningún repositorio hasta que Región de AWS se asocian. Para aplicar una plantilla a uno o varios repositorios, debe asociarla al repositorio o a los repositorios. Puede aplicar una misma plantilla a varios repositorios de una Región de AWS. Esto le ayuda a automatizar y estandarizar el flujo de trabajo de desarrollo en sus repositorios mediante la creación de condiciones coherentes para aprobar y combinar solicitudes de extracción.

Solo puede asociar una plantilla de regla de aprobación a repositorios de la Región de AWS donde se creó la plantilla de regla de aprobación.

Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Temas

- [Asociar una plantilla de regla de aprobación \(consola\)](#)
- [Asociar una plantilla de regla de aprobación \(AWS CLI\)](#)

Asociar una plantilla de regla de aprobación (consola)

Es posible que haya asociado repositorios con una plantilla de regla de aprobación al crearla. Este paso es opcional. Puede añadir o quitar asociaciones editando la plantilla.

Para asociar una plantilla de regla de aprobación con repositorios

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Seleccione Approval rule templates (Plantillas de reglas de aprobación). Elija la plantilla y, a continuación, elija Edit (Editar).
3. En Associated Repositories (Repositorios asociados), elija los repositorios en la lista Repositories (Repositorios). Cada repositorio asociado aparece debajo del cuadro de lista.
4. Seleccione Save. Las reglas de aprobación se aplican ahora a las solicitudes de extracción creadas en los repositorios asociados.

Asociar una plantilla de regla de aprobación (AWS CLI)

Puede utilizarla AWS CLI para asociar una plantilla de reglas de aprobación a uno o más repositorios.

Para asociar una plantilla con un único repositorio

1. En el terminal o en la línea de comandos, ejecute el comando `associate-approval-rule-template-with-repository` especificando lo siguiente:
 - Nombre de la plantilla de regla de aprobación que desea asociar con un repositorio.
 - Nombre del repositorio que se va a asociar con la plantilla de regla de aprobación.

Por ejemplo, para asociar una plantilla de reglas de aprobación denominada **2-approver-rule-for-main** a un repositorio denominado **MyDemoRepo**:

```
aws codecommit associate-approval-rule-template-with-repository --repository-name MyDemoRepo --approval-rule-template-name 2-approver-rule-for-main
```

2. Si se ejecuta correctamente, este comando no devuelve nada.

Para asociar una plantilla con varios repositorios

1. En el terminal o en la línea de comandos, ejecute el comando `batch-associate-approval-rule-template-with-repositories` especificando lo siguiente:
 - Nombre de la plantilla de regla de aprobación que desea asociar con un repositorio.
 - Los nombres de los repositorios que se asociarán con la plantilla de regla de aprobación.

Por ejemplo, para asociar una plantilla de regla de aprobación denominada **2-approver-rule-for-main** con dos repositorios denominados **MyDemoRepo** y **MyOtherDemoRepo**:

```
aws codecommit batch-associate-approval-rule-template-with-repositories --repository-names "MyDemoRepo", "MyOtherDemoRepo" --approval-rule-template-name 2-approver-rule-for-main
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "associatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Administrar plantillas de reglas de aprobación

Puede gestionar las plantillas de reglas de aprobación de forma individual Región de AWS para comprender cómo se utilizan y para qué sirven. Por ejemplo, puede editar los nombres y las descripciones de las plantillas de reglas de aprobación para ayudar a otros a comprender su propósito. Puede enumerar todas las plantillas de reglas de aprobación en una Región de AWS plantilla y obtener información sobre el contenido y la estructura de una plantilla. Puede revisar qué plantillas están asociadas con un repositorio y qué repositorios están asociados con una plantilla.

Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Administrar plantillas de reglas de aprobación (consola)

Puede ver y gestionar sus plantillas de reglas de aprobación en la CodeCommit consola.

Para administrar plantillas de reglas de aprobación

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Seleccione Plantillas de reglas de aprobación para ver la lista de plantillas de reglas de aprobación en la Región de AWS que ha iniciado sesión.

Note

Las plantillas de reglas de aprobación solo están disponibles en el Región de AWS lugar donde se crearon.

3. Si desea realizar cambios en una plantilla, selecciónela en la lista y, a continuación, elija Edit (Editar).

4. Realice sus cambios y, a continuación, elija Save.

Administrar plantillas de reglas de aprobación (AWS CLI)

Puede administrar las plantillas de reglas de aprobación con los siguientes comandos de la AWS CLI :

- [list-approval-rule-templates](#), para ver una lista de todas las plantillas de reglas de aprobación de un Región de AWS
- [get-approval-rule-template](#), para ver el contenido de una plantilla de regla de aprobación
- [update-approval-rule-template-content](#), para cambiar el contenido de una plantilla de regla de aprobación
- [update-approval-rule-template-name](#), para cambiar el nombre de una plantilla de regla de aprobación
- [update-approval-rule-template-description](#), para cambiar la descripción de una plantilla de regla de aprobación
- [list-repositories-for-approval-rule-template](#), para ver todos los repositorios asociados con una plantilla de regla de aprobación
- [list-associated-approval-rule-templates-for-repository](#), para ver todas las plantillas de reglas de aprobación asociadas con un repositorio

Para enumerar todas las plantillas de reglas de aprobación en un Región de AWS

1. En el terminal o la línea de comandos, ejecute el comando `list-approval-rule-templates`. Por ejemplo, para enumerar todas las plantillas de reglas de aprobación de la región Este de EE. UU. (Ohio):

```
aws codecommit list-approval-rule-templates --region us-east-2
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

```
}
```

Obtención del contenido de una plantilla de regla de aprobación

1. En la terminal o en la línea de comandos, ejecute el comando `get-approval-rule-template`, especificando el nombre de la plantilla de regla de aprobación:

```
aws codecommit get-approval-rule-template --approval-rule-template-name 1-approver-
rule-for-all-pull-requests
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements
\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers
\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}\",
    "ruleContentSha256": "621181bbEXAMPLE",
    "lastModifiedDate": 1571356106.936,
    "creationDate": 1571356106.936,
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan",
    "approvalRuleTemplateId": "a29abb15-EXAMPLE",
    "approvalRuleTemplateDescription": "All pull requests must be approved by
one developer on the team."
  }
}
```

Actualización del contenido de una plantilla de regla de aprobación

1. En el terminal o en el símbolo del sistema, ejecute el comando `update-approval-rule-template-content`, especificando el nombre de la plantilla y el contenido modificado. Por ejemplo, para cambiar el contenido de una plantilla de regla de aprobación denominada **1-approver-rule** con el fin de redefinir el grupo de aprobación de manera que incluya a los usuarios que asuman el rol **CodeCommitReview**:

```
aws codecommit update-approval-rule-template-content --approval-rule-template-
name 1-approver-rule --new-rule-content "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type
```

```
\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]]]}]"]
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull
requests from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{ \"Version\": \"2018-11-08\", \"Statements
\": [{ \"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers
\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"] } ] }\",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}
```

Actualización del nombre de una plantilla de regla de aprobación

1. En el terminal o en el símbolo del sistema, ejecute el comando `update-approval-rule-template-name`, especificando el nombre actual y el nombre por el que desea cambiarlo. Por ejemplo, para cambiar el nombre de una plantilla de regla de aprobación de **1-approver-rule** a **1-approver-rule-for-all-pull-requests**:

```
aws codecommit update-approval-rule-template-name --old-approval-rule-template-name
"1-approver-rule" --new-approval-rule-template-name "1-approver-rule-for-all-pull-
requests"
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
```

```

    "approvalRuleTemplateContent": "{\n\"Version\": \"2018-11-08\", \"Statements\n\": [{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\n\": [\n\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]\n}}]",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "All pull requests must be approved by\none developer on the team.",
    "ruleContentSha256": "2f6c21a5cEXAMPLE"
  }
}

```

Actualización de la descripción de una plantilla de regla de aprobación

1. En la terminal o en la línea de comandos, ejecute el comando `update-approval-rule-template-description`, especificando el nombre de la plantilla de regla de aprobación y la nueva descripción:

```

aws codecommit update-approval-rule-template-description --approval-rule-template-
name "1-approver-rule-for-all-pull-requests" --approval-rule-template-description
"Requires 1 approval for all pull requests from the CodeCommitReview pool"

```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```

{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull\nrequests from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\n\"Version\": \"2018-11-08\", \"Statements\n\": [{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\n\": [\n\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]\n}}]",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}

```

Para enumerar todos los repositorios asociados con una plantilla

1. En la terminal o en la línea de comandos, ejecute el comando `list-repositories-for-approval-rule-template`, especificando el nombre de la plantilla:

```
aws codecommit list-repositories-for-approval-rule-template --approval-rule-template-name 2-approver-rule-for-main
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

Para enumerar todas las plantillas asociadas con un repositorio

1. En la terminal o en la línea de comandos, ejecute el comando `list-associated-approval-rule-templates-for-repository`, especificando el nombre del repositorio:

```
aws codecommit list-associated-approval-rule-templates-for-repository --repository-name MyDemoRepo
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Desasociar una plantilla de regla de aprobación

Si las reglas de aprobación generadas por una plantilla de regla de aprobación ya no tienen sentido para el flujo de trabajo del equipo en un repositorio, puede desasociar la plantilla de ese repositorio.

Al desasociar una plantilla, no se eliminan las reglas de aprobación creadas mientras la plantilla estaba asociada al repositorio.

Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Desasociar una plantilla de regla de aprobación (consola)

Puede utilizar la consola para eliminar la asociación entre un repositorio y una plantilla de regla de aprobación.

Para desasociar una plantilla de regla de aprobación de los repositorios

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Seleccione Approval rule templates (Plantillas de reglas de aprobación). Elija la plantilla que desea desasociar de uno o más repositorios y, a continuación, elija Edit (Editar).
3. En Associated repositories (Repositorios asociados), elija la X junto a los repositorios que desea desasociar. Los nombres de los repositorios ya no aparecen.
4. Seleccione Save. Las reglas de aprobación no se aplican a las solicitudes de extracción creadas en esos repositorios. Las reglas continuarán aplicándose a las solicitudes de extracción realizadas mientras la asociación estaba activa.

Desasociar una plantilla de regla de aprobación (AWS CLI)

Puede utilizarla AWS CLI para desasociar uno o más repositorios de una plantilla de reglas de aprobación.

Desasociación de una plantilla de regla de aprobación de un repositorio

1. En el terminal o en la línea de comandos, ejecute el comando `disassociate-approval-rule-template-from-repository` especificando lo siguiente:
 - El nombre de la plantilla de regla de aprobación.
 - El nombre del repositorio.

Por ejemplo, para desasociar una plantilla de regla de aprobación denominada **1-approver-rule-for-all-pull-requests** de un repositorio denominado **MyDemoRepo**:

```
aws codecommit disassociate-approval-rule-template-from-repository --repository-name MyDemoRepo --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. Si se ejecuta correctamente, este comando no devuelve nada.

Para desasociar una plantilla de regla de aprobación de varios repositorios

1. En el terminal o en la línea de comandos, ejecute el comando `batch-disassociate-approval-rule-template-from-repositories` especificando lo siguiente:
 - El nombre de la plantilla de regla de aprobación.
 - Los nombres de los repositorios.

Por ejemplo, para desasociar una plantilla de regla de aprobación denominada **1-approver-rule-for-all-pull-requests** de un repositorio denominado **MyDemoRepo** y de un repositorio denominado **MyOtherDemoRepo**:

```
aws codecommit batch-disassociate-approval-rule-template-from-repositories --repository-names "MyDemoRepo", "MyOtherDemoRepo" --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "disassociatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Eliminar una plantilla de regla de aprobación

Puede eliminar una plantilla de regla de aprobación si no la está utilizando en ningún repositorio. La eliminación de plantillas de reglas de aprobación no utilizadas ayuda a mantener las plantillas organizadas y facilita la búsqueda de plantillas que tengan sentido para los flujos de trabajo.

Para obtener más información sobre las políticas administradas y los permisos para las plantillas de reglas de aprobación, consulte [Permisos para acciones en plantillas de reglas de aprobación](#) y [AWS políticas gestionadas para CodeCommit](#).

Temas

- [Eliminar una plantilla de regla de aprobación \(consola\)](#)
- [Eliminar una plantilla de regla de aprobación \(AWS CLI\)](#)

Eliminar una plantilla de regla de aprobación (consola)

Puede eliminar una plantilla de regla de aprobación si ya no es pertinente para el trabajo de desarrollo. Cuando se utiliza la consola para eliminar una plantilla de regla de aprobación, se desasocia de los repositorios durante el proceso de eliminación.

Eliminación de una plantilla de regla de aprobación

1. [Abre la CodeCommit consola en codecommit/home. `https://console.aws.amazon.com/codesuite/`](#)
2. Seleccione Approval rule templates (Plantillas de reglas de aprobación). Elija la plantilla que desea eliminar y, a continuación, elija Delete (Eliminar).

Eliminar una plantilla de regla de aprobación (AWS CLI)

Puede utilizarla AWS CLI para eliminar una regla de aprobación si se ha disociado de todos los repositorios. Para obtener más información, consulte [Desasociar una plantilla de regla de aprobación \(AWS CLI\)](#).

Eliminación de una plantilla de regla de aprobación

1. En la terminal o en la línea de comandos, ejecute el comando `delete-approval-rule-template`, especificando el nombre de la plantilla de regla de aprobación que desea eliminar:

```
aws codecommit delete-approval-rule-template --approval-rule-template-name 1-  
approver-for-all-pull-requests
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente. Si la plantilla de regla de aprobación ya se ha eliminado, este comando no devuelve nada.

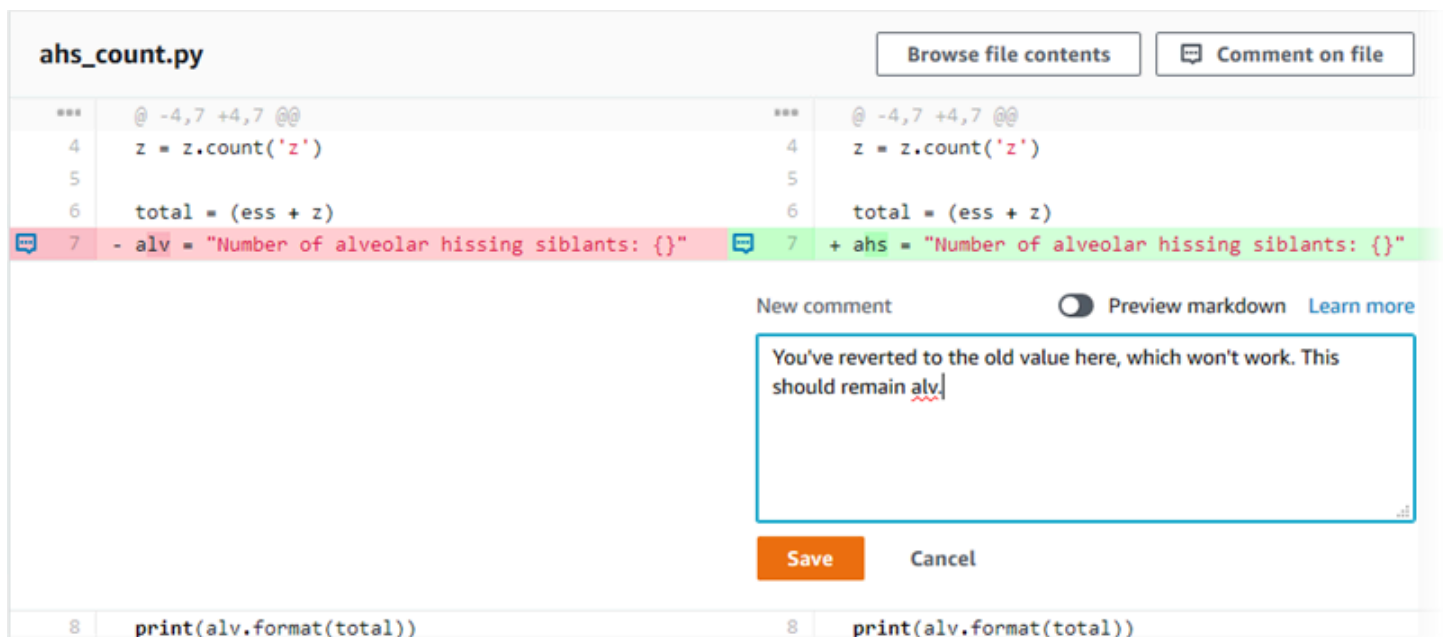
```
{
```

```
"approvalRuleTemplateId": "41de97b7-EXAMPLE"  
}
```

Trabajar con confirmaciones en AWS CodeCommit repositorios

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, dicha información se guarda y se almacena. Esta información incluye datos sobre la persona que confirmó el cambio, la fecha y la hora de la confirmación y los cambios realizados como parte de la confirmación. También se pueden añadir etiquetas a las confirmaciones para identificar fácilmente confirmaciones específicas. En CodeCommit, puedes:

- Revisar confirmaciones
- Ver el historial de confirmaciones en un gráfico
- Comparar una confirmación con su elemento principal o con otro especificador
- Añadir comentarios a sus confirmaciones y responder a los comentarios de otras personas



Para poder enviar las confirmaciones a un CodeCommit repositorio, debes configurar tu ordenador local para que se conecte al repositorio. Para un método más sencillo, consulte [Para usuarios HTTPS mediante credenciales de Git](#).

Para obtener información sobre cómo trabajar con otros aspectos del repositorio en CodeCommit [Trabajar con repositorios](#), consulte [Trabajar con archivos](#) [Trabajo con solicitudes de extracción](#), [Uso de las ramificaciones](#), y [Trabajo con las preferencias del usuario](#).

Temas

- [Crea una confirmación en AWS CodeCommit](#)
- [Consulta los detalles de la confirmación en AWS CodeCommit](#)
- [Compara las confirmaciones en AWS CodeCommit](#)
- [Comenta una confirmación en AWS CodeCommit](#)
- [Crea una etiqueta Git en AWS CodeCommit](#)
- [Ver detalles de la etiqueta Git en AWS CodeCommit](#)
- [Eliminar una etiqueta de Git en AWS CodeCommit](#)

Crea una confirmación en AWS CodeCommit

Al crear la primera confirmación para un repositorio nuevo, se utiliza el put-file comando AWS CLI and. Esto crea la primera confirmación y le permite crear y especificar la ramificación predeterminada para su nuevo repositorio. Puedes usar Git o el AWS CLI para crear una confirmación en un CodeCommit repositorio. Si el repositorio local está conectado a un CodeCommit repositorio, usas Git para enviar la confirmación del repositorio local al CodeCommit repositorio. Para crear una confirmación directamente en la CodeCommit consola, consulta [Crear o añadir un archivo a un repositorio de AWS CodeCommit](#) y. [Editar el contenido de un archivo en un AWS CodeCommit repositorio](#)

Note

Como práctica recomendada, te recomendamos que utilices las últimas versiones compatibles de AWS CLI Git y otros programas. Si usas el AWS CLI, asegúrate de tener instalada una versión reciente para asegurarte de que estás usando una versión que contenga el create-commit comando.

Temas

- [Cree la primera confirmación para un repositorio mediante el AWS CLI](#)
- [Creación de una confirmación mediante un cliente de Git](#)

- [Crea una confirmación mediante el AWS CLI](#)

Cree la primera confirmación para un repositorio mediante el AWS CLI

Puedes usar el `put-file` comando AWS CLI and the para crear tu primera confirmación para un repositorio. Al usar `put-file`, se crea una primera confirmación que añade un archivo a su repositorio vacío y se crea una ramificación con el nombre que especifique. Designa la nueva ramificación como la ramificación predeterminada de su repositorio.

Note

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para crear la primera confirmación de un repositorio mediante el AWS CLI

1. En su ordenador local, cree el archivo que desee añadir como primer archivo al CodeCommit repositorio. Una práctica habitual es crear un archivo de anotaciones de `README.md` que explique el propósito de este repositorio a los demás usuarios del repositorio. Si incluye un `README.md` archivo, el contenido del archivo se muestra automáticamente en la parte inferior de la página de códigos de su repositorio en la CodeCommit consola.
2. En el terminal o en la línea de comandos, ejecute el comando `put-file` especificando lo siguiente:
 - El nombre del repositorio en el que desea añadir el primer archivo.
 - El nombre de la ramificación que quiere crear como ramificación predeterminada.
 - La ubicación local del archivo. La sintaxis usada para esta ubicación varía dependiendo de su sistema operativo local.
 - El nombre del archivo que desea añadir, incluida la ruta en la que se almacena el archivo actualizado en el repositorio.
 - El nombre de usuario y el correo electrónico que desea asociar a este archivo.
 - Un mensaje de confirmación en el que se explica por qué añadió este archivo.

El nombre de usuario, la dirección de correo electrónico y el mensaje de confirmación son opcionales, pero ayudan a otros usuarios a saber quién realizó el cambio y por qué. Si no

proporciona un nombre de usuario, se utilizará de CodeCommit forma predeterminada su nombre de usuario de IAM o una derivación del inicio de sesión de la consola como nombre de autor.

Por ejemplo, para añadir un archivo *README.md* con el nombre de un ejemplo de contenido de archivo codificado en base 6 a un repositorio con el nombre de una *MyDemoRepo* rama denominada: *development*

```
aws codecommit put-file --repository-name MyDemoRepo --branch-name development --
file-path README.md --file-content "EXAMPLE0123456789example0123456789EXAMPLE1"
--name "Mary Major" --email "mary_major@example.com" --commit-message "I added a
quick readme for our new team repository."
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "commitId": "724caa36EXAMPLE",
  "blobId": "a8a94062EXAMPLE",
  "treeId": "08b2fc73EXAMPLE"
}
```

Creación de una confirmación mediante un cliente de Git

Puedes crear confirmaciones mediante un cliente Git instalado en tu ordenador local y, a continuación, enviar esas confirmaciones a tu CodeCommit repositorio.

1. Cumpla los requisitos previos, incluida la [Configuración](#)

Important

Si no ha completado la instalación, no puede conectarse al repositorio ni realizar la confirmación utilizando Git.

2. Asegúrese de que está creando una confirmación en la ramificación correcta. Para ver una lista de las ramificaciones disponibles y saber cuál de ellas está configurada actualmente para utilizarse, ejecute `git branch`. Se muestran todas las ramificaciones. Un asterisco (*) aparece junto al nombre de la ramificación actual. Para cambiar a otra ramificación, ejecute `git checkout branch-name`. Si esta es su primera confirmación, ejecute el comando `git`

`config` para configurar su cliente Git y crear una ramificación inicial con el nombre que quiera usar para esa ramificación. Por ejemplo, si quieres que tu rama predeterminada tenga el nombre *development*:

```
git config --local init.defaultBranch development
```

 Tip

Este comando solamente está disponible en la versión Git v.2.28 o versiones posteriores.

También puede ejecutar este comando para establecer como nombre de ramificación predeterminado todos los repositorios de **development** recién creados:

```
git config --global init.defaultBranch development
```

3. Realice un cambio en la ramificación (como, por ejemplo, añadir, modificar o eliminar un archivo).

Por ejemplo, en el repositorio local, cree un archivo denominado `bird.txt` con el siguiente texto:

```
bird.txt
-----
Birds (class Aves or clade Avialae) are feathered, winged, two-legged, warm-blooded, egg-laying vertebrates.
```

4. Ejecute `git status`, que debería indicar que `bird.txt` aún no se ha incluido en ninguna confirmación pendiente:

```
...
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    bird.txt
```

5. Ejecute `git add bird.txt` para incluir el nuevo archivo en la confirmación pendiente.
6. Si vuelve a ejecutar `git status`, debería ver un resultado similar a este. Indica que `bird.txt` ahora forma parte de la confirmación pendiente o está preparado para confirmarse:

```
...
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   bird.txt
```

7. Para finalizar la confirmación, ejecute `git commit` con la opción `-m` (por ejemplo, `git commit -m "Adding bird.txt to the repository."`). La opción `-m` crea el mensaje de confirmación.
8. Si vuelve a ejecutar `git status`, debería ver un resultado similar a este. Indica que la confirmación está lista para enviarse del repositorio local al CodeCommit repositorio:

```
...
nothing to commit, working directory clean
```

9. Antes de enviar la confirmación finalizada desde el repositorio local al CodeCommit repositorio, puedes ver qué es lo que estás insertando. Este `remote-name` es el apodo que el repositorio local usa para el CodeCommit repositorio y el nombre de la rama que `branch-name` se va a comparar. `git diff --stat remote-name/branch-name`

Tip

Para obtener el alias, ejecute `git remote`. Para obtener una lista de los nombres de las ramificaciones, ejecute `git branch`. Un asterisco (*) aparece junto al nombre de la ramificación actual. También puede ejecutar `git status` para obtener el nombre de la ramificación actual.

Note

Si has clonado el repositorio, desde la perspectiva del repositorio local, no `remote-name` es el nombre del repositorio. CodeCommit Al clonar un repositorio, `remote-name` se establece automáticamente en. `origin`

Por ejemplo, el resultado de `git diff --stat origin/main` sería similar al siguiente:

```
bird.txt | 1 +
```

```
1 file changed, 1 insertion(+)
```

El resultado supone que ya has conectado el repositorio local al CodeCommit repositorio. (Para obtener instrucciones, consulte [Conectarse a un repositorio](#)).

10. Cuando estés listo para enviar la confirmación del repositorio local al CodeCommit repositorio, ejecuta `git push remote-name branch-name`, que **remote-name** es el apodo que el repositorio local usa para el CodeCommit repositorio y el nombre de la rama que **branch-name** se va a enviar al repositorio. CodeCommit

Por ejemplo, el resultado de ejecutar `git push origin main` sería similar al siguiente:

Para HTTPS:

```
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 516 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
    b9e7aa6..3dbf4dd main -> main
```

Para SSH:

```
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 516 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote:
To ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
    b9e7aa6..3dbf4dd main -> main
```

Tip

Si añade la opción `-u` a `git push`(por ejemplo, `git push -u origin main`), en el futuro solamente tendrá que ejecutar `git push`, ya que la información de seguimiento ascendente queda configurada. Para obtener información de seguimiento ascendente, ejecute `git remote show remote-name` (por ejemplo, `git remote show origin`).

Para ver más opciones, consulte la documentación de Git.

Crea una confirmación mediante el AWS CLI

Puedes usar el `create-commit` comando AWS CLI and para crear una confirmación para un repositorio en la punta de una rama específica. También puede crear una confirmación de combinación sin referencia para representar los resultados de la combinación de dos especificadores de confirmación. Para obtener más información, consulte [Crear una confirmación sin referencia](#).

Note

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Creación de una confirmación

1. En su equipo local, realice los cambios que desee confirmar en repositorio de CodeCommit.
2. En el terminal o en la línea de comandos, ejecute el comando `create-commit` especificando lo siguiente:
 - El repositorio en el que desea confirmar los cambios.
 - La ramificación en la que desea confirmar los cambios.
 - El ID de confirmación completo de la confirmación más reciente realizada a esa ramificación, también conocido como confirmación TIP o HEAD o el ID de confirmación principal.
 - Si se mantienen las carpetas vacías si los cambios que ha realizado eliminan el contenido de estas carpetas. De forma predeterminada, este valor es `false`.
 - La información acerca de los archivos que desea añadir, cambiar o eliminar.
 - El nombre de usuario y el correo electrónico que desea asociar a estos cambios.
 - Un mensaje de confirmación que explica por qué ha realizado estos cambios.

El nombre de usuario, la dirección de correo electrónico y el mensaje de confirmación son opcionales, pero ayudan a otros usuarios a saber quién realizó los cambios y por qué. Si no proporciona un nombre de usuario, se utilizará de CodeCommit forma predeterminada su nombre de usuario de IAM o una derivación del inicio de sesión de la consola como nombre de autor.

Por ejemplo, para crear una confirmación para un repositorio que añada un README.md archivo a un repositorio con el nombre *MyDemoRepo* de la rama *main*. El contenido del archivo está en Base64 y dice "¡Bienvenido al repositorio de nuestro equipo!":

```
aws codecommit create-commit --repository-name MyDemoRepo --  
branch-name main --parent-commit-id 4c925148EXAMPLE --put-files  
"filePath=README.md,fileContent=V2VsY29tZSB0byBvdXIgdGVhbSBvZXBvc2l0b3J5IQo="
```

 Tip

Para obtener el ID de confirmación principal, ejecute el comando [get-branch](#).

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{  
  "commitId": "4df8b524-EXAMPLE",  
  "treeId": "55b57003-EXAMPLE",  
  "filesAdded": [  
    {  
      "blobId": "5e1c309dEXAMPLE",  
      "absolutePath": "meeting.md",  
      "fileMode": "NORMAL"  
    }  
  ],  
  "filesDeleted": [],  
  "filesUpdated": []  
}
```

Para crear una confirmación que realice cambios en los archivos denominados *file1.py* y *file2.txt*, cambie el nombre de *picture.png* a un archivo *image1.png* y lo mueva de un directorio denominado *pictures* a un directorio denominado *images*, y elimine un archivo nombrado *ExampleSolution.py* en un repositorio denominado *MyDemoRepo* en una rama denominada *MyFeatureBranch* cuya confirmación más reciente tenga un identificador de: *4c925148EXAMPLE*

```
aws codecommit create-commit --repository-name MyDemoRepo --branch-  
name MyFeatureBranch --parent-commit-id 4c925148EXAMPLE --author-name "Saanvi  
Sarkar"
```

```
--email "saanvi_sarkar@example.com" --commit-message "I'm creating this commit to
update a variable name in a number of files."
--keep-empty-folders false --put-files '{"filePath": "file1.py", "fileMode":
"EXECUTABLE", "fileContent": "bucket_name = sys.argv[1] region = sys.argv[2]}"'
'{"filePath": "file2.txt", "fileMode": "NORMAL", "fileContent": "//Adding a comment
to explain the variable changes in file1.py"}' '{"filePath": "images/image1.png",
"fileMode": "NORMAL", "sourceFile": {"filePath": "pictures/picture.png", "isMove":
true}}' --delete-files filePath="ExampleSolution.py"
```

Note

La sintaxis para el segmento `--put-files` varía en función de su sistema operativo. En el ejemplo anterior está optimizado para usuarios de Linux, macOS, o Unix y usuarios de Windows con un emulador de Bash. Los usuarios de Windows en la línea de comandos o en PowerShell deberían utilizar una sintaxis adecuada para estos sistemas.

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE",
  "filesAdded": [
    {
      "absolutePath": "images/image1.png",
      "blobId": "d68ba6ccEXAMPLE",
      "fileMode": "NORMAL"
    }
  ],
  "filesUpdated": [
    {
      "absolutePath": "file1.py",
      "blobId": "0a4d55a8EXAMPLE",
      "fileMode": "EXECUTABLE"
    },
    {
      "absolutePath": "file2.txt",
      "blobId": "915766bbEXAMPLE",
      "fileMode": "NORMAL"
    }
  ],
}
```

```
"filesDeleted": [  
  {  
    "absolutePath": "ExampleSolution.py",  
    "blobId": "4f9cebe6aEXAMPLE",  
    "fileMode": "EXECUTABLE"  
  },  
  {  
    "absolutePath": "pictures/picture.png",  
    "blobId": "fb12a539EXAMPLE",  
    "fileMode": "NORMAL"  
  }  
]  
}
```

Consulta los detalles de la confirmación en AWS CodeCommit

Puedes usar la AWS CodeCommit consola para explorar el historial de confirmaciones de un repositorio. Esto puede ayudarte a identificar los cambios realizados en un repositorio, entre los que se incluyen:

- El autor y el momento en que se han realizado los cambios
- El momento en que se han combinado confirmaciones específicas en una ramificación.

Ver el historial de confirmaciones de una ramificación también puede ayudar a comprender la diferencia entre las ramificaciones. Si utiliza etiquetas, también podrá ver rápidamente la confirmación etiquetada con una etiqueta y los elementos principales de dicha confirmación etiquetada. En la línea de comandos, puedes usar Git para ver detalles sobre las confirmaciones en un repositorio o CodeCommit repositorio local.

Examinación de confirmaciones de un repositorio

Puedes usar la AWS CodeCommit consola para explorar el historial de confirmaciones en un repositorio. También puede ver un gráfico de las confirmaciones del repositorio y sus ramificaciones a lo largo del tiempo. Esto puede ayudarte a comprender el historial del repositorio, incluido el momento en que se efectúan cambios.

 Note

Si utiliza el comando git rebase para cambiar la base de un repositorio, modificará el historial de un repositorio, lo que puede provocar que las confirmaciones aparezcan desordenadas. Para obtener más información, consulta [Git Branching-Rebasing](#) o la documentación de Git.

Temas

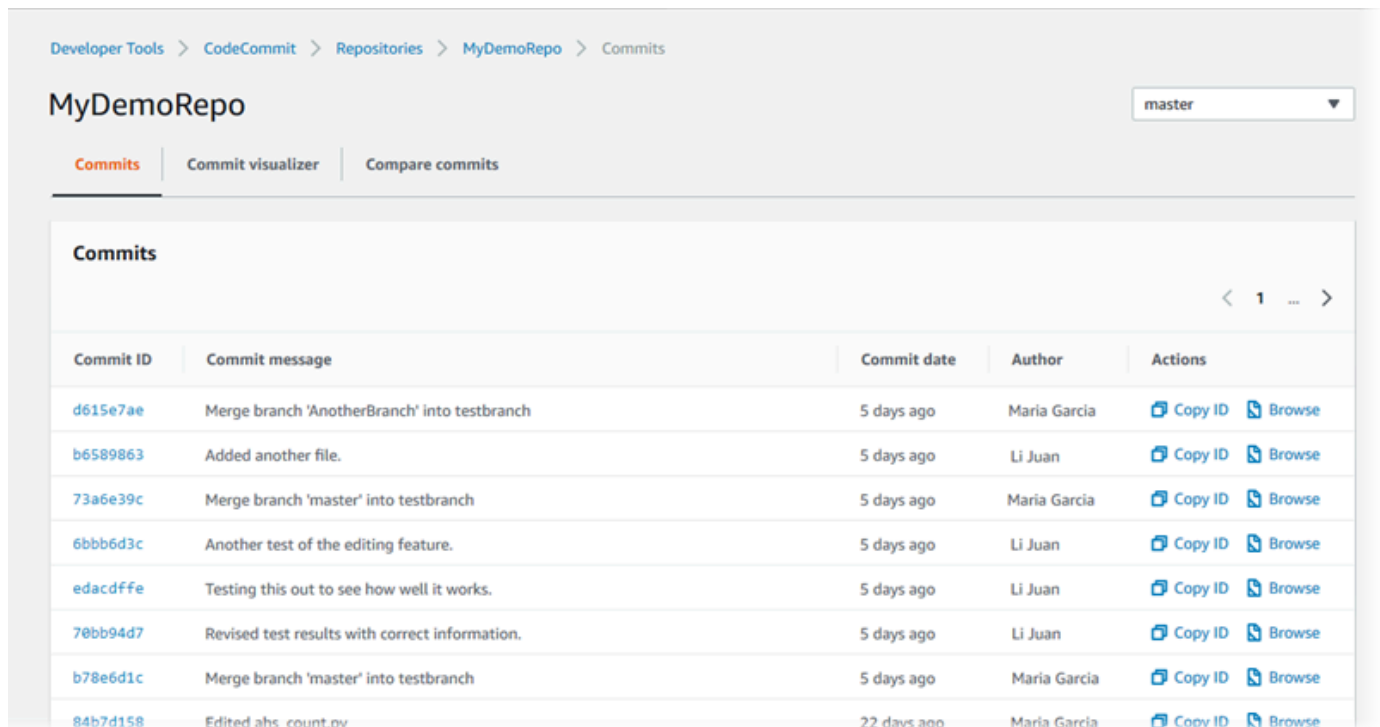
- [Examinación del historial de confirmaciones de un repositorio](#)
- [Visualización de un gráfico del historial de confirmaciones de un repositorio](#)

Examinación del historial de confirmaciones de un repositorio

Puede examinar el historial de confirmaciones de una ramificación específica o etiqueta del repositorio, incluida información sobre la herramienta de confirmación y sobre el mensaje de confirmación. También puede ver el código de una confirmación.

Para examinar el historial de confirmaciones

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio para el que desea revisar el historial de confirmaciones.
3. En el panel de navegación, elija Commits. En la vista del historial de confirmaciones se muestra un historial de confirmaciones del repositorio de la ramificación predeterminada en orden cronológico inverso a la fecha de la confirmación. La fecha y la hora se muestran según la hora universal coordinada (UTC). Puede ver el historial de confirmaciones de una ramificación diferente eligiendo el botón de selección de vista y, a continuación, una ramificación de la lista. Si utiliza etiquetas en el repositorio, puede ver una confirmación con una etiqueta específica y sus elementos principales marcando dicha etiqueta en el botón de selección de vista.



4. Para ver la diferencia entre una confirmación y su principal, y para ver los comentarios sobre los cambios, elija el ID de la confirmación abreviado. Para obtener más información, consulte [Comparación de una confirmación con su elemento principal](#) y [Comentar una confirmación](#). Para ver la diferencia entre una confirmación y cualquier otro especificador de confirmación, incluida una ramificación, etiqueta o ID de confirmación, consulte [Comparación de dos especificadores de confirmación](#).
5. Realice una o más de las siguientes acciones:
 - Para ver la fecha y la hora de un cambio, pase el cursor sobre la fecha de confirmación.
 - Para ver el ID completo de la confirmación, copie y péguelo en un editor de texto o en otra ubicación. Para copiarlo, elija Copy ID (Copiar ID).
 - Para ver el código como aparecía en el momento de una confirmación, elija Browse (Examinar). En la vista Code, se muestra el contenido del repositorio en el momento de dicha confirmación. El botón de selección de vista muestra el ID de confirmación abreviado en lugar de una ramificación o etiqueta.

Visualización de un gráfico del historial de confirmaciones de un repositorio

Puede ver un gráfico de las confirmaciones realizadas en un repositorio. La vista Commit Visualizer es una representación de gráfico acíclico dirigido (DAG) de todas las confirmaciones realizadas en

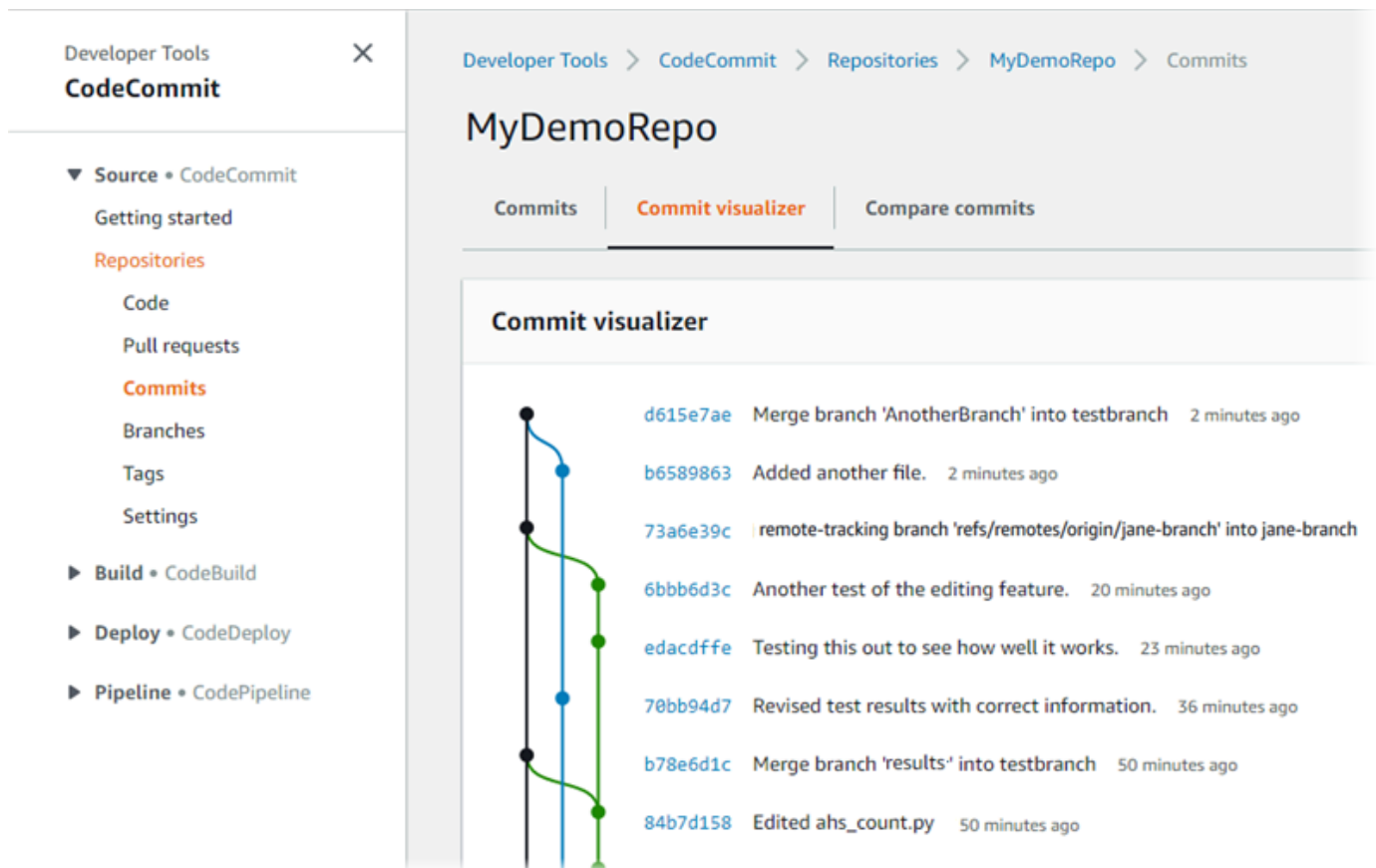
una ramificación del repositorio. Esta representación gráfica puede ayudarle a entender cuándo se añadieron o combinaron las confirmaciones y sus características asociadas. También puede ayudarle a identificar cuándo se realizó un cambio en relación con otros cambios.

Note

Las confirmaciones que se combinan mediante el método fast-forward (de avance rápido) no aparecen como líneas separadas en el gráfico de confirmaciones.

Para ver un gráfico de las confirmaciones

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio para el que desea ver un gráfico de confirmación.
3. En el panel de navegación, seleccione Commits (Confirmaciones) y, a continuación, seleccione la pestaña Commit visualizer (Confirmar visualizador).



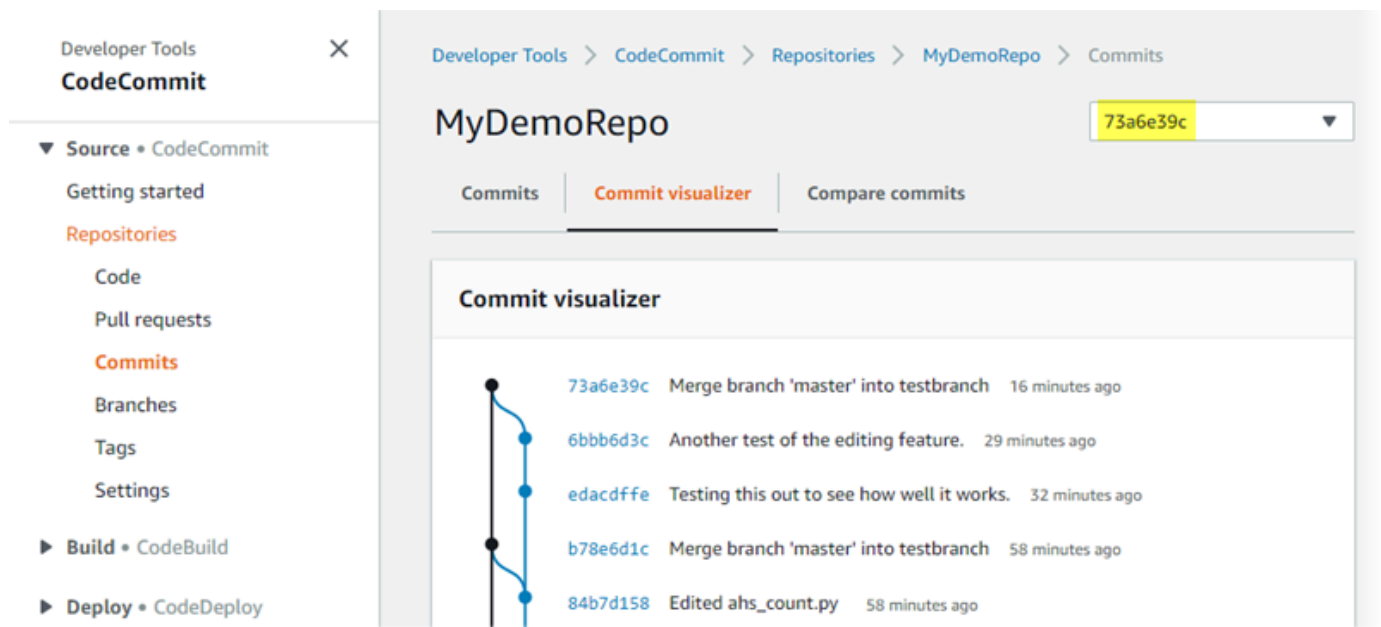
En el gráfico de confirmaciones, el ID de confirmación abreviado y el asunto de cada mensaje de confirmación aparece junto a dicho punto en el gráfico.

Note

El gráfico puede mostrar hasta 35 ramificaciones en una página. Si hay más de 35 ramificaciones, el gráfico será demasiado complejo para mostrarlo. Puede simplificar la vista de dos formas:

- Usando el botón selector de vista para mostrar el gráfico de una ramificación específica
- Pegando un ID completo de la confirmación en el cuadro de búsqueda para representar el gráfico de dicha confirmación

4. Para representar un nuevo gráfico de una confirmación, elija el punto en el gráfico que corresponde a dicha confirmación. El botón de selección de vista cambia al ID de confirmación abreviado.



Ver los detalles de la confirmación (AWS CLI)

Git le permite ver la información detallada de las confirmaciones. También puedes usarlo AWS CLI para ver los detalles de las confirmaciones en un repositorio local o en un CodeCommit repositorio ejecutando los siguientes comandos:

- Para ver la información sobre una confirmación, ejecute [aws codecommit get-commit](#).
- Para ver información acerca de varias confirmaciones, ejecute [aws codecommit batch-get-commits](#).
- Para ver información sobre una confirmación de combinación, ejecute [aws codecommit get-merge-commit](#).
- Ejecute [aws codecommit get-differences](#) para ver información acerca de los cambios de un especificador de confirmación (ramificación, etiqueta, HEAD u otras referencias completas, como los ID de confirmación).
- Ejecute [aws codecommit get-blob](#) para ver el contenido con codificación base64 de un objeto blob de Git en un repositorio.

Para ver información sobre una confirmación

1. Ejecute el comando `aws codecommit get-commit`, especificando:

- El nombre del CodeCommit repositorio (con la `--repository-name` opción).
- El ID completo de la confirmación

Por ejemplo, para ver la información sobre una confirmación con el ID 317f8570EXAMPLE en un CodeCommit repositorio llamado MyDemoRepo:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id
317f8570EXAMPLE
```

2. Si se ejecuta correctamente, el resultado de este comando incluye lo siguiente:
 - Información sobre el autor de la confirmación (según la configuración en Git), incluida la fecha en formato de marca de hora y la diferencia respecto al tiempo universal coordinado (UTC)
 - Información sobre el confirmador (según la configuración en Git), incluida la fecha en formato de marca de hora y la diferencia respecto a la UTC
 - El ID del árbol Git donde está la confirmación
 - El ID de la confirmación principal
 - El mensaje de la confirmación

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "commit": {
    "additionalData": "",
    "committer": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "author": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "treeId": "347a3408EXAMPLE",
    "parents": [
      "4c925148EXAMPLE"
    ],
    "message": "Fix incorrect variable name"
  }
}
```

Para ver información sobre una confirmación de combinación

1. Ejecute el comando `get-merge-commit`, especificando:

- Un especificador de confirmación para el origen de la combinación (con la opción `--source-commit-specifier`).
- Un especificador de confirmación para el destino para la combinación (con la opción `--destination-commit-specifier`).
- La opción de combinación que desea utilizar (con la opción `--merge-option`).
- El nombre del repositorio (con la opción `--repository-name`).

Por ejemplo, para ver información sobre una confirmación de fusión para la rama de origen nombrada *bugfix-bug1234* con una rama de destino denominada *main* mediante la *THREE_WAY_MERGE* estrategia en un repositorio denominado *MyDemoRepo*:

```
aws codecommit get-merge-commit --source-commit-specifier bugfix-bug1234 --  
destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-  
name MyDemoRepo
```

2. Si se ejecuta correctamente, el resultado de este comando proporciona información similar a la siguiente:

```
{  
  "sourceCommitId": "c5709475EXAMPLE",  
  "destinationCommitId": "317f8570EXAMPLE",  
  "baseCommitId": "fb12a539EXAMPLE",  
  "mergeCommitId": "ffc4d608eEXAMPLE"  
}
```

Visualización de información acerca de varias confirmaciones

1. Ejecute el comando `batch-get-commits`, especificando:
 - El nombre del CodeCommit repositorio (con la `--repository-name` opción).
 - Una lista de ID de confirmación completos para cada confirmación de la que desea ver información.

Por ejemplo, para ver la información sobre las confirmaciones con los ID `317f8570EXAMPLE` y `4c925148EXAMPLE` en un CodeCommit repositorio llamado `MyDemoRepo`:

```
aws codecommit batch-get-commits --repository-name MyDemoRepo --commit-ids  
317f8570EXAMPLE 4c925148EXAMPLE
```

2. Si se ejecuta correctamente, el resultado de este comando incluye lo siguiente:
 - Información sobre los autores de las confirmaciones (según la configuración en Git), incluida la fecha en formato de marca de hora y la diferencia respecto al tiempo universal coordinado (UTC)
 - Información sobre los confirmadores (según la configuración en Git), incluida la fecha en formato de marca de hora y la diferencia respecto a la UTC
 - Los ID del árbol Git donde está la confirmación.
 - Los ID de confirmación principal.

- Los mensajes de la confirmación

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "commits": [
    {
      "additionalData": "",
      "committer": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "author": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "commitId": "317f8570EXAMPLE",
      "treeId": "1f330709EXAMPLE",
      "parents": [
        "6e147360EXAMPLE"
      ],
      "message": "Change variable name and add new response element"
    },
    {
      "additionalData": "",
      "committer": {
        "date": "1508280542 -0800",
        "name": "Li Juan",
        "email": "li_juan@example.com"
      },
      "author": {
        "date": "1508280542 -0800",
        "name": "Li Juan",
        "email": "li_juan@example.com"
      },
      "commitId": "4c925148EXAMPLE",
      "treeId": "1f330709EXAMPLE",
      "parents": [
        "317f8570EXAMPLE"
      ],
    },
  ]
}
```

```
    "message": "Added new class"
  }
}
```

Para ver información acerca de los cambios de un especificador de confirmación

1. Ejecute el comando `aws codecommit get-differences`, especificando:

- El nombre del CodeCommit repositorio (con la `--repository-name` opción).
- Los especificadores de confirmación sobre los que desea obtener información. Solo se requiere `--after-commit-specifier`. Si no especifica `--before-commit-specifier`, se muestran todos los archivos actuales de `--after-commit-specifier`

Por ejemplo, para ver información sobre las diferencias entre las confirmaciones con los ID 317f8570EXAMPLE y 4c925148EXAMPLE las de un CodeCommit repositorio llamado MyDemoRepo:

```
aws codecommit get-differences --repository-name MyDemoRepo --before-commit-specifier 317f8570EXAMPLE --after-commit-specifier 4c925148EXAMPLE
```

2. Si se ejecuta correctamente, el resultado de este comando incluye lo siguiente:

- Una lista de diferencias, incluido el tipo de cambio (A = añadido, D = eliminado y M = modificado)
- El mecanismo del archivo de tipo de cambio
- El ID del objeto blob Git que contiene el cambio

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3bEXAMPLE",
        "mode": "100644"
      },
      "changeType": "M",
    }
  ]
}
```



```
        "beforeBlob": {
            "path": "blob.txt",
            "blobId": "bf7fcf28fEXAMPLE",
            "mode": "100644"
        }
    ]
}
```

Visualización de información sobre un objeto blob de Git

1. Ejecute el comando `aws codecommit get-blob`, especificando:

- El nombre del CodeCommit repositorio (con la `--repository-name` opción).
- El ID del blob Git (con la opción `--blob-id`)

Por ejemplo, para ver información sobre un blob de Git con el ID de `2eb4af3bEXAMPLE` en un CodeCommit repositorio llamado `MyDemoRepo`:

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

2. Si se ejecuta correctamente, el resultado de este comando incluye lo siguiente:

- El contenido con codificación base64 del blob, normalmente un archivo

Por ejemplo, el resultado del comando anterior puede ser similar al siguiente:

```
{
  "content": "QSBcCaW5hcnkgTGFGyToEXAMPLE="
}
```

Visualización de la información detallada de una confirmación (Git)

Antes de seguir estos pasos, deberías haber conectado el repositorio local al CodeCommit repositorio y haber confirmado los cambios. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

Para mostrar los cambios de la confirmación más reciente de un repositorio, ejecute el comando `git show`.

```
git show
```

El resultado del comando debería ser similar al siguiente:

```
commit 4f8c6f9d
Author: Mary Major <mary.major@example.com>
Date:   Mon May 23 15:56:48 2016 -0700

    Added bumblebee.txt

diff --git a/bumblebee.txt b/bumblebee.txt
new file mode 100644
index 00000000..443b974
--- /dev/null
+++ b/bumblebee.txt
@@ -0,0 +1 @@
+A bumblebee, also written bumble bee, is a member of the bee genus Bombus, in the
  family Apidae.
\ No newline at end of file
```

Note

En este y los siguientes ejemplos, los ID de la confirmación se han acortado. El ID completo de la confirmación no se muestra.

Para ver los cambios que se han producido, utilice el comando `git show` con el ID de confirmación:

```
git show 94ba1e60

commit 94ba1e60
Author: John Doe <johndoe@example.com>
Date:   Mon May 23 15:39:14 2016 -0700

    Added horse.txt

diff --git a/horse.txt b/horse.txt
new file mode 100644
```

```
index 00000000..080f68f
--- /dev/null
+++ b/horse.txt
@@ -0,0 +1 @@
+The horse (Equus ferus caballus) is one of two extant subspecies of Equus ferus.
```

Para ver las diferencias entre dos confirmaciones, ejecute el comando `git diff` e incluya los dos ID de la confirmación.

```
git diff ce22850d 4f8c6f9d
```

En este ejemplo, la diferencia entre las dos confirmaciones es que se han añadido dos archivos. El resultado del comando debería ser similar al siguiente:

```
diff --git a/bees.txt b/bees.txt
new file mode 100644
index 00000000..cf57550
--- /dev/null
+++ b/bees.txt
@@ -0,0 +1 @@
+Bees are flying insects closely related to wasps and ants, and are known for their
  role in pollination and for producing honey and beeswax.
diff --git a/bumblebee.txt b/bumblebee.txt
new file mode 100644
index 00000000..443b974
--- /dev/null
+++ b/bumblebee.txt
@@ -0,0 +1 @@
+A bumblebee, also written bumble bee, is a member of the bee genus Bombus, in the
  family Apidae.
\ No newline at end of file
```

Para utilizar Git para ver información detallada acerca de las confirmaciones en un repositorio local, ejecute el comando `git log`:

```
git log
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
commit 94ba1e60
Author: John Doe <johndoe@example.com>
```

```
Date:   Mon May 23 15:39:14 2016 -0700
```

```
    Added horse.txt
```

```
commit 4c925148
```

```
Author: Jane Doe <janedoe@example.com>
```

```
Date:   Mon May 22 14:54:55 2014 -0700
```

```
    Added cat.txt and dog.txt
```

Para mostrar solo ID de la confirmación y mensajes, ejecute el comando `git log --pretty=oneline`:

```
git log --pretty=oneline
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
94ba1e60 Added horse.txt
```

```
4c925148 Added cat.txt and dog.txt
```

Para ver más opciones, consulte la documentación de Git.

Compara las confirmaciones en AWS CodeCommit

Puedes usar la CodeCommit consola para ver las diferencias entre los especificadores de confirmación de un CodeCommit repositorio. Puede ver rápidamente la diferencia entre una confirmación y su elemento primario. También puede comparar dos referencias, incluidos los ID de las confirmaciones.

Temas

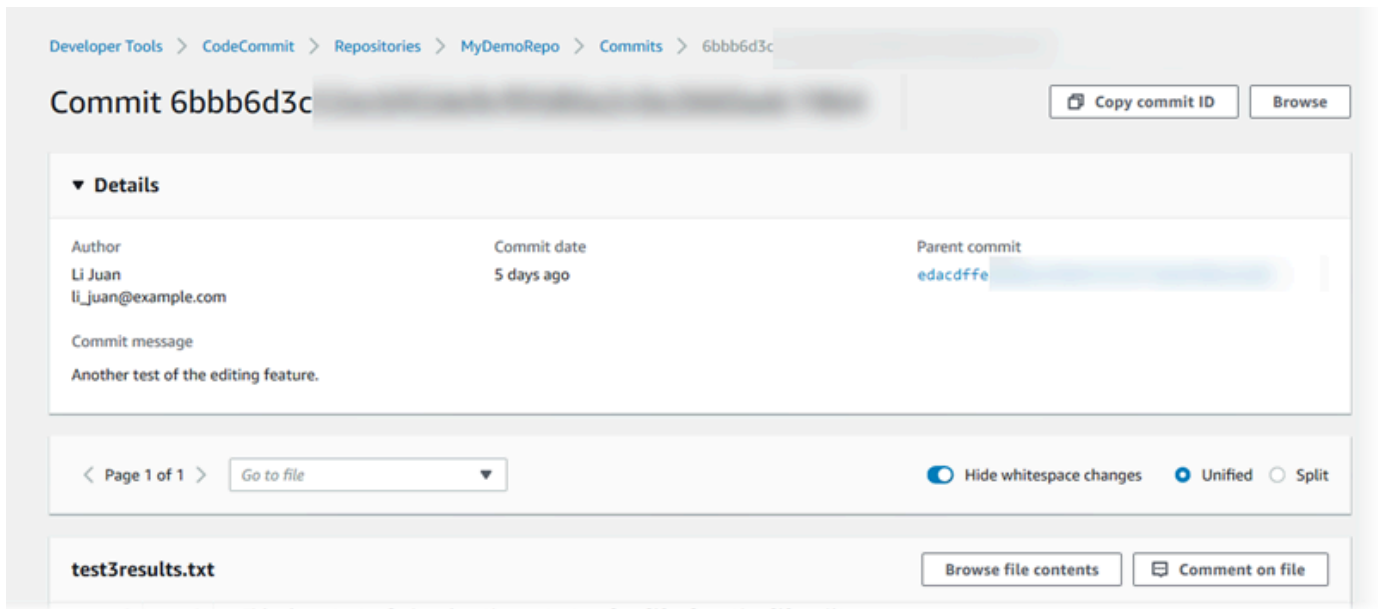
- [Comparación de una confirmación con su elemento principal](#)
- [Comparación de dos especificadores de confirmación](#)

Comparación de una confirmación con su elemento principal

Puede ver rápidamente la diferencia entre una confirmación y su elemento primario para revisar el mensaje de confirmación, quién confirma y qué es lo que ha cambiado.

1. Abre la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>

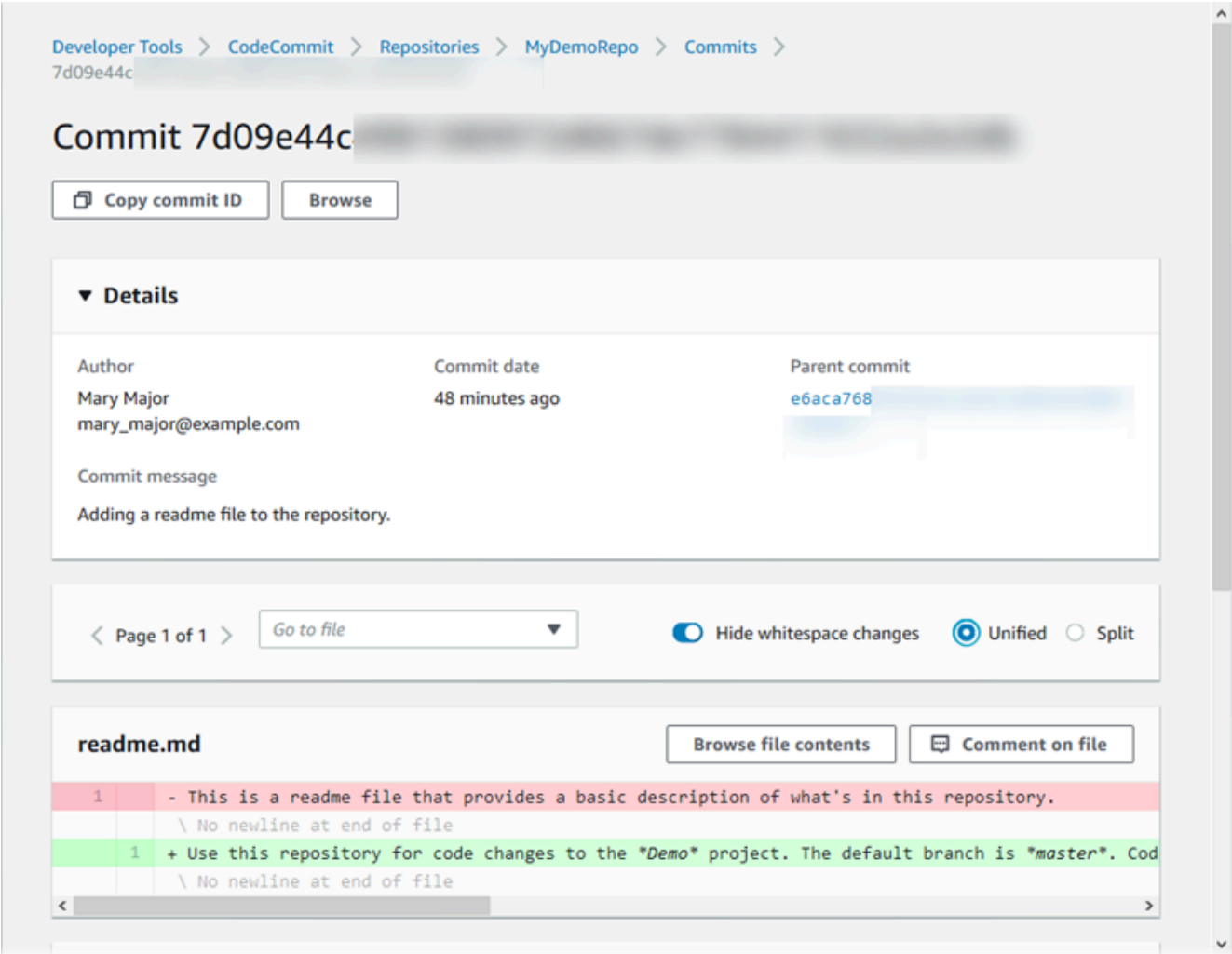
2. En la página Repositories (Repositorios), elija el repositorio en el que desea ver la diferencia entre una confirmación y su elemento principal.
3. En el panel de navegación, elija Commits.
4. Elija el ID de confirmación abreviado de cualquier confirmación de la lista. La vista cambia para mostrar la información de esa confirmación, incluidas las diferencias entre ella y su elemento primario.

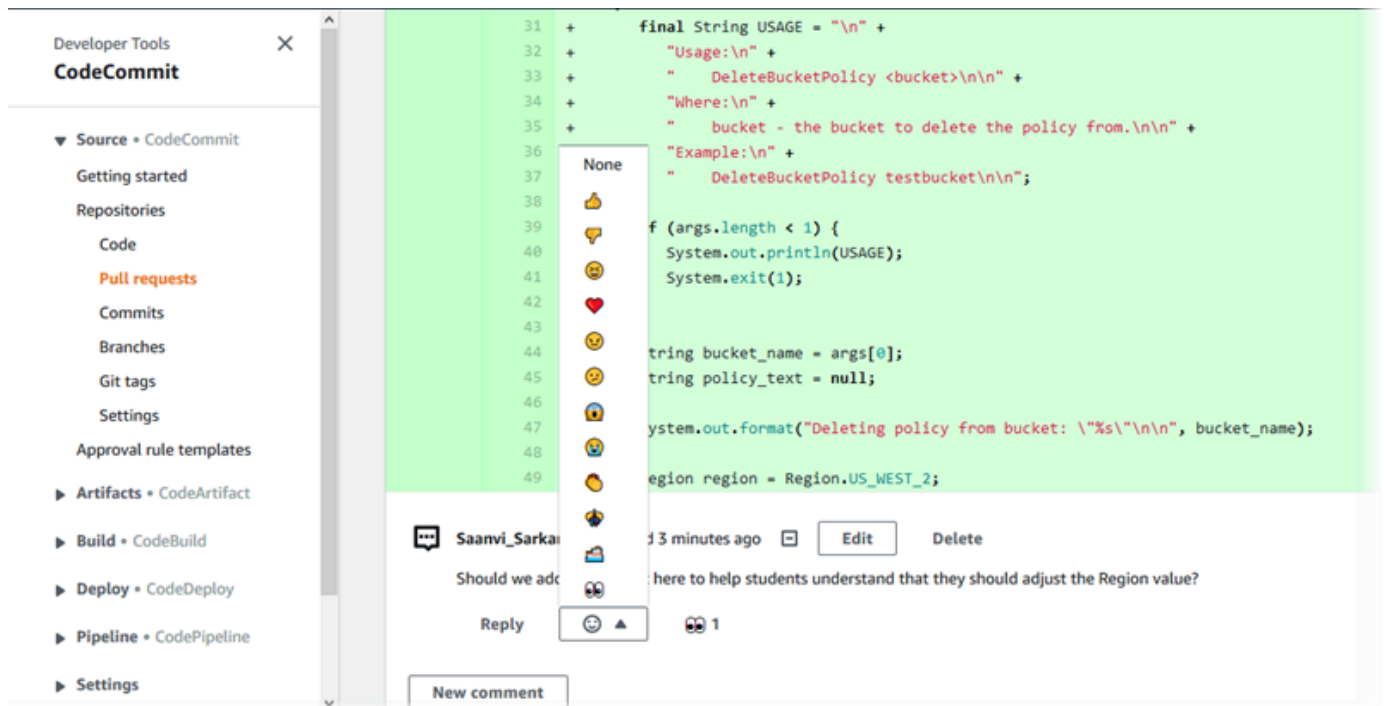


Puede mostrar los cambios de forma paralela (vista Split) o en línea (vista Unified). También puede mostrar u ocultar los cambios en los espacios en blanco. También puede añadir comentarios. Para obtener más información, consulte [Comentar una confirmación](#).

Note

Las preferencias para ver el código y otra configuración de la consola se guardan como cookies del navegador siempre que se modifican. Para obtener más información, consulte [Trabajo con las preferencias del usuario](#).





Note

En función del estilo de fin de línea, del editor de código y de otros factores, es posible que vea líneas completas añadidas o eliminadas en lugar de cambios concretos en una línea. El nivel de detalle coincide con lo que devuelven los comandos `git show` o `git diff`.

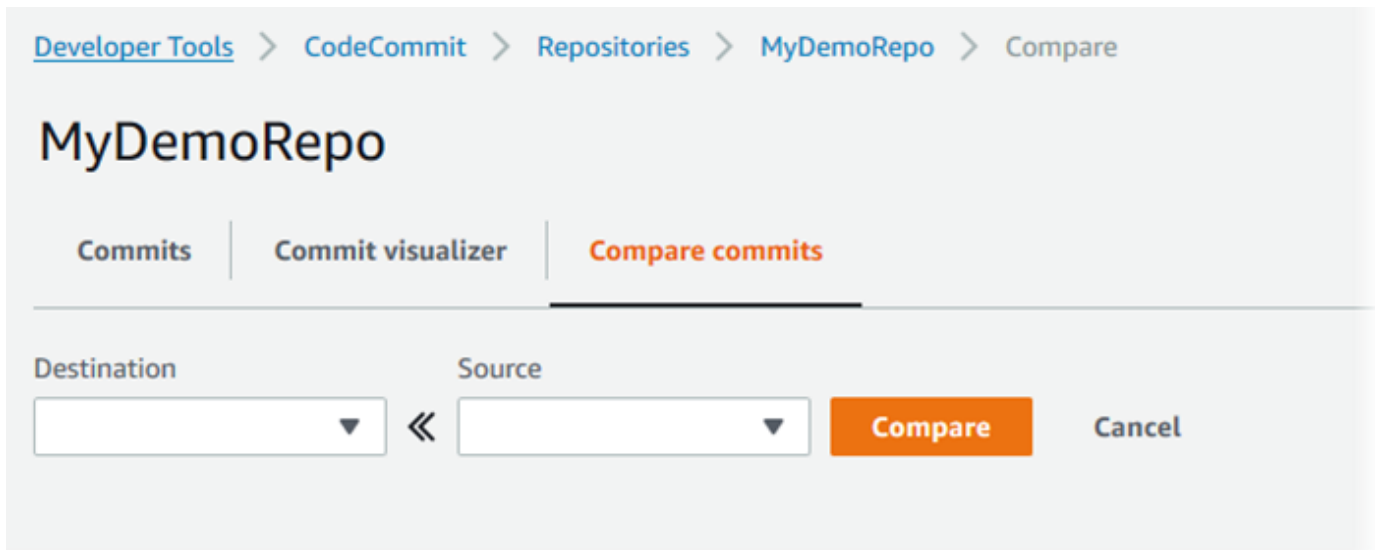
- Para comparar una confirmación con su elemento principal desde Commit visualizer (Visualizador de confirmación), seleccione el ID de confirmación abreviado. Se muestran los detalles de la confirmación, incluidos los cambios entre esta y su elemento principal.

Comparación de dos especificadores de confirmación

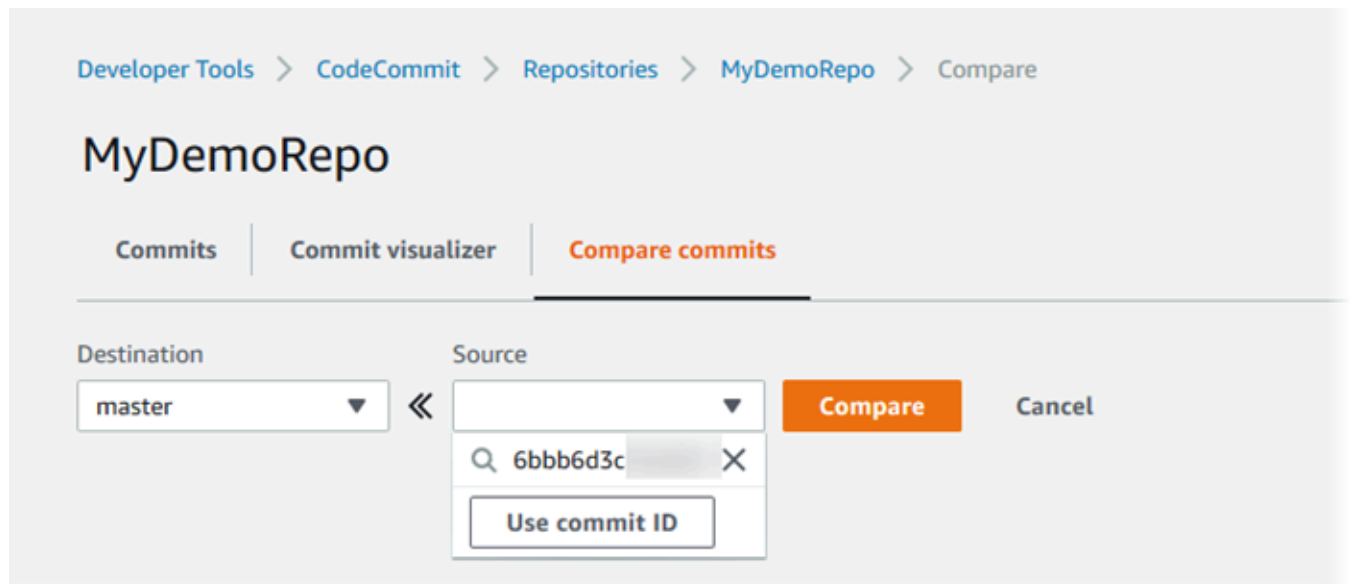
Puedes ver las diferencias entre dos especificadores de confirmación en la CodeCommit consola. Los especificadores de confirmación son referencias como, por ejemplo, ramificaciones, etiquetas e ID de confirmación.

- Abre la CodeCommit consola en. <https://console.aws.amazon.com/codesuite/codecommit/home>
- En la página Repositories (Repositorios), seleccione el repositorio cuyas confirmaciones, ramificaciones o confirmaciones etiquetadas desea comparar.

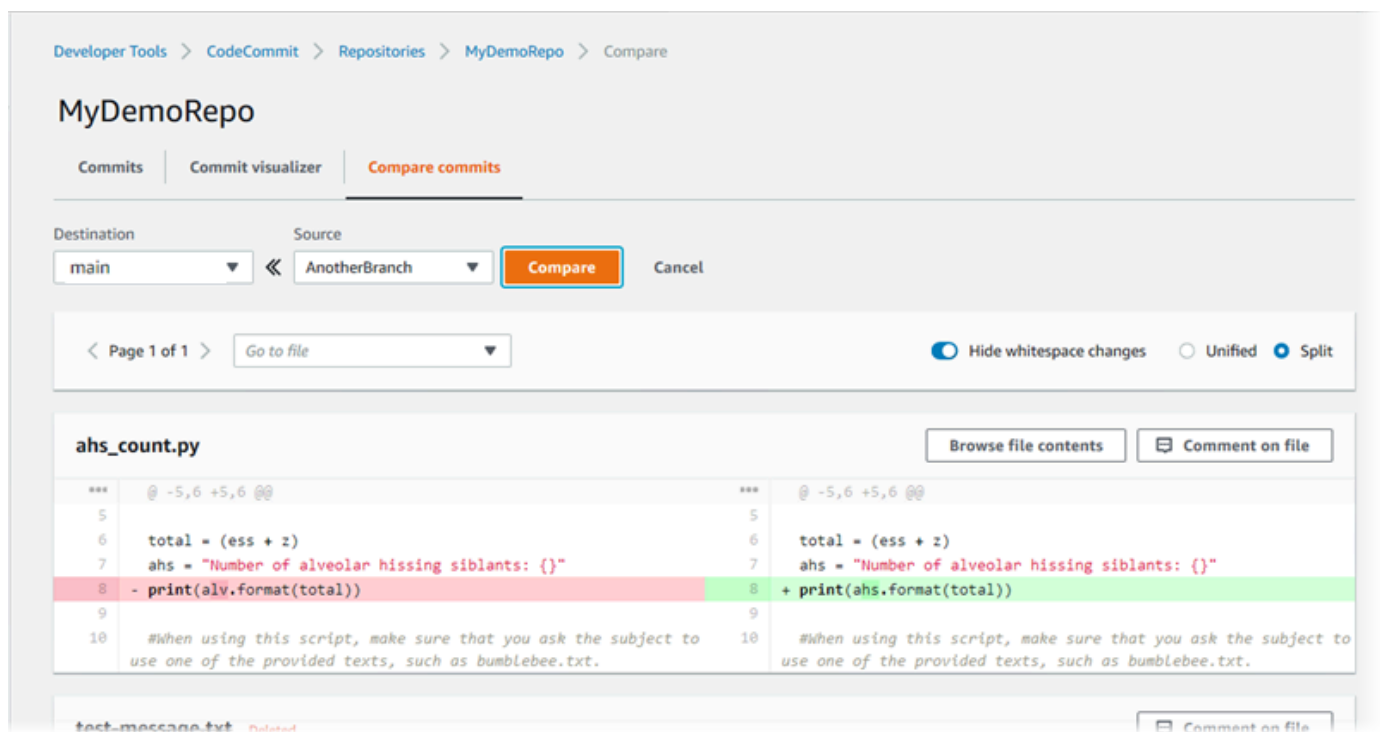
3. En el panel de navegación, seleccione Commits (Confirmaciones) y haga clic en Compare commits (Comparar confirmaciones).



4. Utilice los cuadros para comparar dos especificadores de confirmación.
 - Para comparar el extremo de una ramificación, seleccione el nombre de la ramificación en la lista. Esto selecciona, para la comparación, la confirmación más reciente de esa ramificación.
 - Para comparar una confirmación con una etiqueta específica asociada a ella, seleccione el nombre de la etiqueta en la lista, si hay alguna. Esto selecciona la confirmación etiquetada que se va a comparar.
 - Para comparar una confirmación específica, introduzca o pegue el ID de la confirmación en el cuadro. Para obtener el ID completo de la confirmación, elija Commits en la barra de navegación y, a continuación, copie el ID de confirmación que aparece en la lista. En la página Compare commits (Comparar confirmaciones), pegue el ID de confirmación completo en el cuadro de texto y seleccione Use commit ID (Usar ID de confirmación).



5. Una vez seleccionados los especificadores, elija Compare.



Puede mostrar las diferencias de forma paralela (vista Split) o en línea (vista Unified). También puede mostrar u ocultar los cambios en los espacios en blanco.

6. Para borrar las opciones de comparación, elija Cancel (Cancelar).

Comenta una confirmación en AWS CodeCommit

Puedes usar la CodeCommit consola para comentar las confirmaciones en un repositorio y ver y responder a los comentarios de otros usuarios sobre las confirmaciones. Esto puede ayudarle a explicar los cambios realizados en un repositorio, entre los que se incluyen:

- Por qué se han efectuado cambios.
- Si se necesitan más cambios.
- Si se deben combinar los cambios en otra ramificación.

Puede realizar un comentario sobre una confirmación en su conjunto, un archivo en de una confirmación o una línea o un cambio específico en un archivo. También puede vincular una línea de código seleccionándola y copiando la URL resultante en el navegador.

Note

Para obtener un resultado óptimo, cree los comentarios cuando haya iniciado sesión como usuario de IAM. La funcionalidad de comentarios no está optimizada para los usuarios que inician sesión con credenciales de cuenta raíz, acceso federado o credenciales temporales.

Temas

- [Ver los comentarios de una confirmación en un repositorio](#)
- [Añadir y responder a los comentarios de una confirmación en un repositorio](#)
- [Ver, añadir, actualizar y responder a los comentarios \(AWS CLI\)](#)

Ver los comentarios de una confirmación en un repositorio

Puedes usar la CodeCommit consola para ver los comentarios de una confirmación.

Visualización de los comentarios sobre una confirmación

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio cuyos comentarios sobre las confirmaciones desea revisar.

3. En el panel de navegación, elija Commits. Seleccione el ID de la confirmación cuyos comentarios desea ver.

Se muestra la página para esa confirmación, junto con los comentarios que tenga.

Añadir y responder a los comentarios de una confirmación en un repositorio

Puede utilizar la CodeCommit consola para añadir comentarios a la comparación de una confirmación y una principal, o a la comparación entre dos confirmaciones especificadas. También puede responder a los comentarios con emojis, con sus propios comentarios o con ambos.

Añadir y responder a los comentarios de una confirmación (consola)

Puede añadir comentarios a una confirmación y responder con texto y emojis. Los comentarios y emojis se marcan como pertenecientes al usuario de IAM o el rol que utilizó para iniciar sesión en la consola.

Par añadir y responder a los comentarios de una confirmación

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el repositorio donde desea realizar comentarios sobre las confirmaciones.
3. En el panel de navegación, elija Commits. Seleccione el ID de la confirmación en la que desee agregar o responder a un comentario.

Se muestra la página para esa confirmación, junto con los comentarios que tenga.

4. Para añadir un comentario, lleve a cabo alguna de las siguientes operaciones:
 - Para añadir un comentario general, en Comments on changes (Comentar cambios), introduzca el comentario y, a continuación, elija Save (Guardar). Puede utilizar [Markdown \(Marcado\)](#) o introducir un comentario en texto sin formato.

Comments on changes

New comment

☐ Preview markdown

[Learn more](#)

Did we also change the variable name in blf.py and concat.py?

Save

- Para añadir un comentario a un archivo de la confirmación, busque el nombre del archivo. Seleccione Comment on file (Comentario de archivo), introduzca un comentario y elija Save (Guardar).

ahs_count.py

Browse file contents

 Comment on file

New comment

☐ Preview markdown

[Learn more](#)

Save

Cancel

- Para añadir un comentario en una línea modificada de la confirmación, vaya a la línea en la que aparece el cambio. Seleccione el globo de comentario



introduzca un comentario y después elija Save (Guardar).

ahs_count.py [Browse file contents](#) [Comment on file](#)

```
*** @ -4,7 +4,7 @@
4  z = z.count('z')
5
6  total = (ess + z)
7  - alv = "Number of alveolar hissing sibilants: {}"
   + ahs = "Number of alveolar hissing sibilants: {}"
8
8  print(alv.format(total))
```

New comment ☒ Preview markdown [Learn more](#)

You've reverted to the old value here, which won't work. This should remain alv.

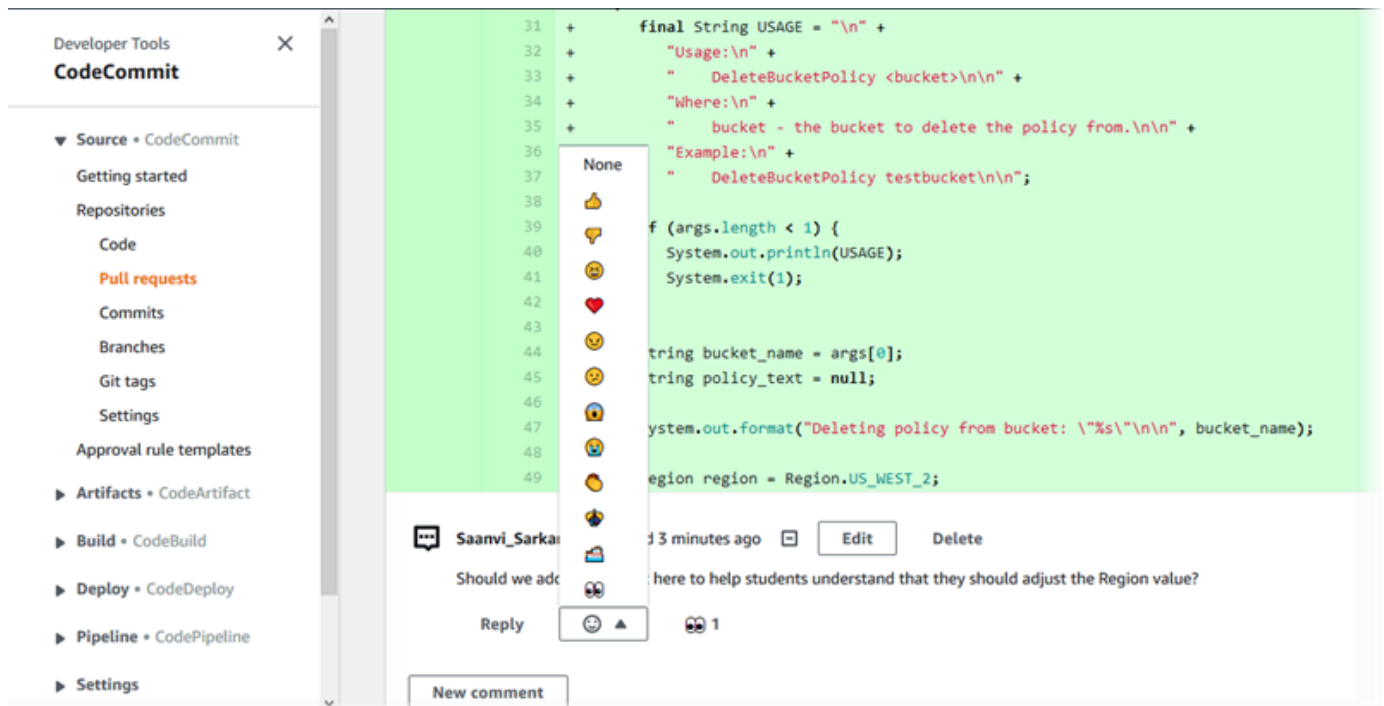
[Save](#) [Cancel](#)

```
8  print(alv.format(total))
```

Note

Puede editar su comentario después de haberlo guardado. También puede eliminar su contenido. El comentario permanecerá con un mensaje que indica que se ha eliminado el contenido. Conviene que utilice el modo Preview markdown (Vista previa de marcado) para el comentario antes de guardarlo.

5. Para responder a los comentarios de una confirmación, seleccione Reply (Responder). Para responder a un comentario con un emoji, seleccione el emoji que quiera de la lista. Solo puede elegir un emoji por comentario. Si quiere cambiar su reacción con los emojis, seleccione otro de la lista o seleccione Ninguno para eliminar su reacción.

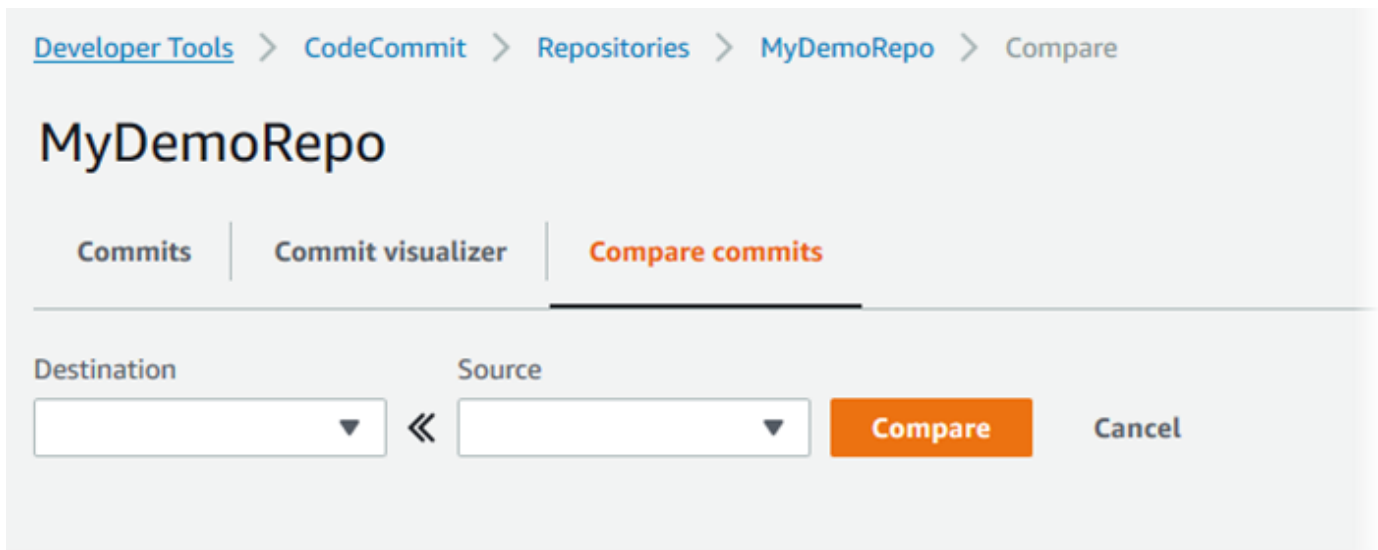


Añadir y responder a los comentarios cuando se comparan dos especificadores de confirmación

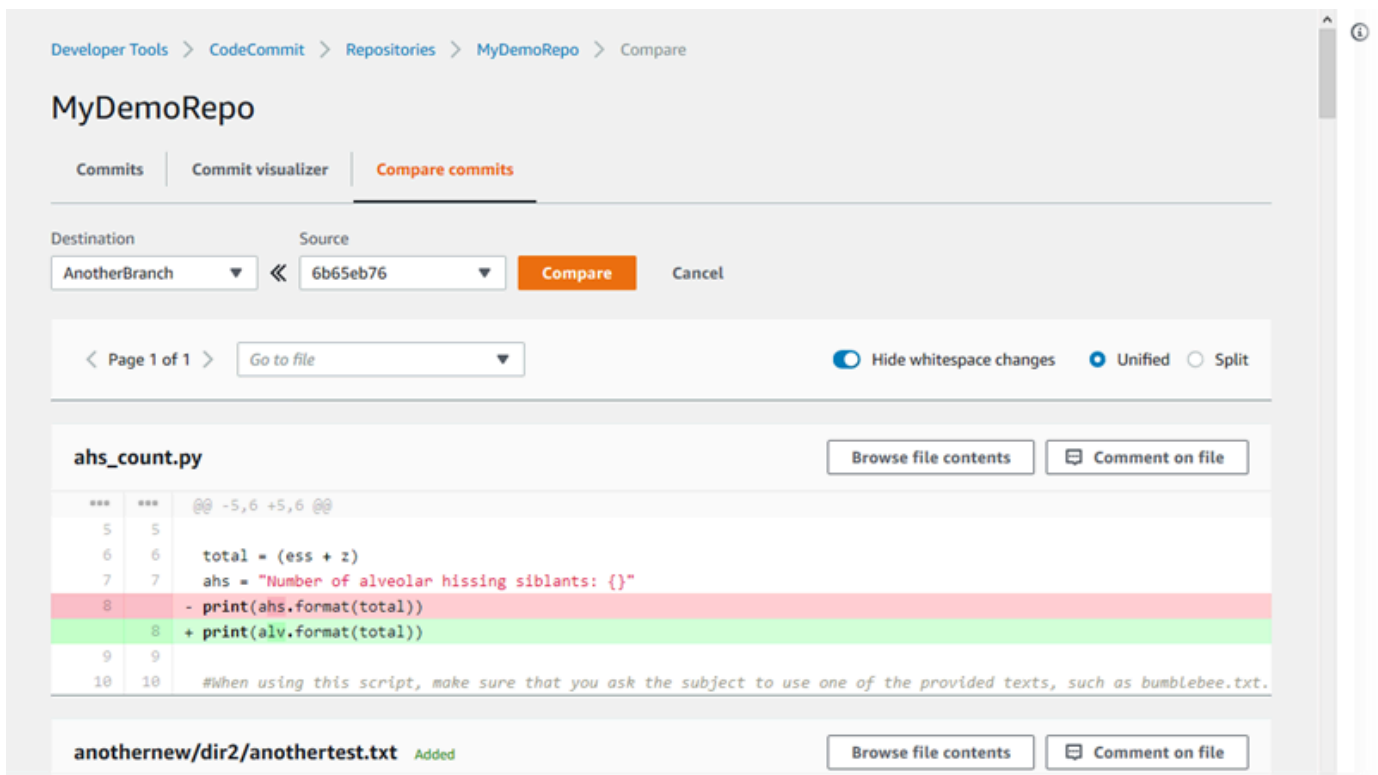
Puede añadir comentarios a una comparación entre ramificaciones, etiquetas o confirmaciones.

Para añadir o responder a los comentarios cuando se comparan dos especificadores de confirmación

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En la página Repositories (Repositorios), elija el repositorio cuyas confirmaciones, ramificaciones o confirmaciones etiquetadas desea comparar.
3. En el panel de navegación, seleccione Commits (Confirmaciones) y haga clic en la pestaña Compare commits (Comparar confirmaciones).



4. Utilice los campos Destination (Destino) y Source (Origen) para comparar dos especificadores de confirmación. Utilice las listas desplegables o pegue los ID de confirmación. Elija Compare.



5. Realice una o más de las siguientes acciones:
 - Para añadir comentarios en los archivos o líneas, elija el globo de comentario



- Para añadir comentarios generales sobre los cambios comparados, vaya a [Comments on changes](#).

Ver, añadir, actualizar y responder a los comentarios (AWS CLI)

Puede ver, añadir, responder, actualizar y eliminar el contenido de un comentario ejecutando los siguientes comandos:

- Ejecute [get-comments-for-compared-commit](#) para ver los comentarios de una comparación de dos confirmaciones.
- Ejecute [get-comment](#) para ver los detalles de un comentario específico.
- Ejecute [delete-comment-content](#) para eliminar el contenido de un comentario que ha creado.
- Ejecute [post-comment-for-compared-commit](#) para crear un comentario sobre la comparación de dos confirmaciones.
- Ejecute [update-comment](#) para actualizar un comentario.
- Para responder a un comentario, ejecute [post-comment-reply](#).
- Para responder a un comentario con un emoji, ejecute [put-comment-reaction](#).
- Para ver las reacciones con emojis a un comentario, ejecute [get-comment-reactions](#).

Visualización de los comentarios sobre una confirmación

1. Ejecute el comando `get-comments-for-compared-commit`, especificando:
 - El nombre del CodeCommit repositorio (con la `--repository-name` opción).
 - El ID completo de la confirmación después, para establecer la direccionalidad de la comparación (con la `--after-commit-id` option).
 - El ID completo de la confirmación antes, para establecer la direccionalidad de la comparación (con la opción `--before-commit-id`).
 - (Opcional) Un token de enumeración para devolver el siguiente lote de resultados (con la opción `--next-token`).
 - (Opcional) Un número entero no negativo para limitar en el número de resultados devueltos (con la opción `--max-results`).

Por ejemplo, para ver los comentarios realizados en la comparación de dos confirmaciones en un repositorio denominado *MyDemoRepo*:

```
aws codecommit get-comments-for-compared-commit --repository-name MyDemoRepo --  
before-commit-id 6e147360EXAMPLE --after-commit-id 317f8570EXAMPLE
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "commentsForComparedCommitData": [  
    {  
      "afterBlobId": "1f330709EXAMPLE",  
      "afterCommitId": "317f8570EXAMPLE",  
      "beforeBlobId": "80906a4cEXAMPLE",  
      "beforeCommitId": "6e147360EXAMPLE",  
      "comments": [  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "123Example",  
          "commentId": "ff30b348EXAMPLEb9aa670f",  
          "content": "Whoops - I meant to add this comment to the line, not  
the file, but I don't see how to delete it.",  
          "creationDate": 1508369768.142,  
          "deleted": false,  
          "CommentId": "123abc-EXAMPLE",  
          "lastModifiedDate": 1508369842.278,  
          "callerReactions": [],  
          "reactionCounts":  
            {  
              "SMILE" : 6,  
              "THUMBSUP" : 1  
            }  
        },  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "123Example",  
          "commentId": "553b509bEXAMPLE56198325",  
          "content": "Can you add a test case for this?",  
          "creationDate": 1508369612.240,  
          "deleted": false,  
          "commentId": "456def-EXAMPLE",
```

```

        "lastModifiedDate": 1508369612.240,
        "callerReactions": [],
        "reactionCounts":
        {
            "THUMBSUP" : 2
        }
    },
    "location": {
        "filePath": "cl_sample.js",
        "filePosition": 1232,
        "relativeFileVersion": "after"
    },
    "repositoryName": "MyDemoRepo"
}
],
"nextToken": "exampleToken"
}

```

Para ver los detalles de un comentario de una confirmación

1. Ejecute el comando `get-comment` especificando el ID de comentario generado por el sistema. Por ejemplo:

```
aws codecommit get-comment --comment-id ff30b348EXAMPLEb9aa670f
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```

{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "123Example",
    "commentId": "ff30b348EXAMPLEb9aa670f",
    "content": "Whoops - I meant to add this comment to the line, but I don't see
how to delete it.",
    "creationDate": 1508369768.142,
    "deleted": false,
    "commentId": "",
    "lastModifiedDate": 1508369842.278,
    "callerReactions": [],
    "reactionCounts":
    {

```

```
        "SMILE" : 6,  
        "THUMBSUP" : 1  
    }  
}  
}
```

Para eliminar el contenido de un comentario en una confirmación

1. Ejecute el comando `delete-comment-content` especificando el ID de comentario generado por el sistema. Por ejemplo:

```
aws codecommit delete-comment-content --comment-id ff30b348EXAMPLEb9aa670f
```

Note

Solo puede eliminar el contenido de un comentario si ha aplicado la `AWSCodeCommitFullAccess` política o si tiene el `DeleteCommentContent` permiso establecido en Permitir.

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "comment": {  
    "creationDate": 1508369768.142,  
    "deleted": true,  
    "lastModifiedDate": 1508369842.278,  
    "clientRequestToken": "123Example",  
    "commentId": "ff30b348EXAMPLEb9aa670f",  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "callerReactions": [],  
    "reactionCounts":  
      {  
        "CLAP" : 1  
      }  
  }  
}
```

Creación de un comentario sobre una confirmación

1. Ejecute el comando `post-comment-for-compared-commit`, especificando:
 - El nombre del CodeCommit repositorio (con la `--repository-name` opción).
 - El ID completo de la confirmación después, para establecer la direccionalidad de la comparación (con la opción `--after-commit-id`).
 - El ID completo de la confirmación antes, para establecer la direccionalidad de la comparación (con la opción `--before-commit-id`).
 - Un token de idempotencia único generado por el cliente (con la opción `--client-request-token`).
 - El contenido del comentario (con la opción `--content`).
 - Una lista con información de la ubicación sobre dónde se puede introducir el comentario, que incluya:
 - El nombre del archivo que se está comparando, incluida la extensión y el subdirectorio, si los hubiese (con el atributo `filePath`).
 - El número de línea del cambio del archivo comparado (con el atributo `filePosition`).
 - Si el comentario sobre el cambio se ha realizado antes o después en la comparación entre la ramificación de origen y la de destino (con el atributo `relativeFileVersion`).

Por ejemplo, para añadir el comentario *"Can you add a test case for this?"* sobre el cambio al *cl_sample.js* archivo en la comparación entre dos confirmaciones de un repositorio llamado *MyDemoRepo*:

```
aws codecommit post-comment-for-compared-commit --repository-name MyDemoRepo
--before-commit-id 317f8570EXAMPLE --after-commit-id 5d036259EXAMPLE --client-
request-token 123Example --content "Can you add a test case for this?" --location
filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "afterBlobId": "1f330709EXAMPLE",
  "afterCommitId": "317f8570EXAMPLE",
  "beforeBlobId": "80906a4cEXAMPLE",
  "beforeCommitId": "6e147360EXAMPLE",
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
```

```

        "clientRequestToken": "",
        "commentId": "553b509bEXAMPLE56198325",
        "content": "Can you add a test case for this?",
        "creationDate": 1508369612.203,
        "deleted": false,
        "commentId": "abc123-EXAMPLE",
        "lastModifiedDate": 1508369612.203,
        "callerReactions": [],
        "reactionCounts": []
    },
    "location": {
        "filePath": "cl_sample.js",
        "filePosition": 1232,
        "relativeFileVersion": "AFTER"
    },
    "repositoryName": "MyDemoRepo"
}

```

Actualización de un comentario sobre una confirmación

1. Ejecute el comando `update-comment`, especificando el ID de comentario generado por el sistema y el contenido para sustituir el contenido existente.

Por ejemplo, para añadir el contenido *"Fixed as requested. I'll update the pull request."* a un comentario con un identificador de *442b498bEXAMPLE5756813*:

```
aws codecommit update-comment --comment-id 442b498bEXAMPLE5756813 --content "Fixed
as requested. I'll update the pull request."
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```

{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "",
    "commentId": "442b498bEXAMPLE5756813",
    "content": "Fixed as requested. I'll update the pull request.",
    "creationDate": 1508369929.783,
    "deleted": false,
    "lastModifiedDate": 1508369929.287,
    "callerReactions": [],

```

```
    "reactionCounts":
      {
        "THUMBSUP" : 2
      }
  }
}
```

Para responder a un comentario sobre una confirmación

1. Para publicar una respuesta a un comentario en una solicitud de extracción, ejecute el comando `post-comment-reply` especificando lo siguiente:
 - El ID generado por el sistema del comentario al que desea responder (con la opción `--in-reply-to`).
 - Un token de idempotencia único generado por el cliente (con la opción `--client-request-token`).
 - El contenido de la respuesta (con la opción `--content`).

Por ejemplo, para añadir la respuesta *"Good catch. I'll remove them."* al comentario con el identificador generado por el sistema de: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --
content "Good catch. I'll remove them." --client-request-token 123Example
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "123Example",
    "commentId": "442b498bEXAMPLE5756813",
    "content": "Good catch. I'll remove them.",
    "creationDate": 1508369829.136,
    "deleted": false,
    "CommentId": "abcd1234EXAMPLEb5678efgh",
    "lastModifiedDate": 150836912.221,
    "callerReactions": [],
    "reactionCounts": []
  }
}
```

Para responder a un comentario sobre una confirmación con un emoji

1. Para responder a un comentario de una solicitud de cambios con un emoji o para cambiar el valor de su reacción ante un emoji, ejecute el comando `put-comment-reaction`, especificando:
 - El ID generado por el sistema del comentario al que desea responder con un emoji.
 - El valor de la reacción que quiere añadir o actualizar. Los valores aceptables incluyen emojis, códigos cortos y valores Unicode compatibles.

Los emojis admiten los siguientes valores: CodeCommit

Emoji	Shortcode	Unicode
#	:thumbsup:	U+1F44D
#	:thumbsdown:	U+1F44E
#	:smile:	U+1F604
♥	:heart:	U+2764
#	:angry:	U+1F620
#	:confused:	U+1F615
#	:scream:	U+1F631
#	:sob:	U+1F62D
#	:clap:	U+1F44F
#	:confetti_ball:	U+1F38A
#	:ship:	U+1F6A2
#	:eyes:	U+1F440
	none	U+0000

Por ejemplo, para añadir el emoji `:thumbsup:` al comentario con el identificador generado por el sistema de: `abcd1234EXAMPLEb5678efgh`

```
aws codecommit put-comment-reaction --comment-id abcd1234EXAMPLEb5678efgh --
reaction-value :thumbsup:
```

2. Si se ejecuta correctamente, este comando no genera ninguna salida.

Para ver las reacciones con emojis a un comentario

1. Para ver las reacciones de los emojis a un comentario, incluidos los usuarios que reaccionaron con esos emojis, ejecute el comando `get-comment-reactions` y especifique el identificador del comentario generado por el sistema.

Por ejemplo, para ver las reacciones de los emojis al comentario con el identificador generado por el sistema de: `abcd1234EXAMPLEb5678efgh`

```
aws codecommit get-comment-reactions --comment-id abcd1234EXAMPLEb5678efgh
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "reactionsForComment": {
    [
      {
        "reaction": {
          "emoji": "#",
          "shortCode": "thumbsup",
          "unicode": "U+1F44D"
        },
        "users": [
          "arn:aws:iam::123456789012:user/Li_Juan",
          "arn:aws:iam::123456789012:user/Mary_Major",
          "arn:aws:iam::123456789012:user/Jorge_Souza"
        ]
      },
      {
        "reaction": {
          "emoji": "#",
          "shortCode": "thumbsdown",
```



```
        "unicode": "U+1F44E"
      },
      "users": [
        "arn:aws:iam::123456789012:user/Nikhil_Jayashankar"
      ]
    },
    {
      "reaction": {
        "emoji": "#",
        "shortCode": "confused",
        "unicode": "U+1F615"
      },
      "users": [
        "arn:aws:iam::123456789012:user/Saanvi_Sarkar"
      ]
    }
  ]
}
```

Crea una etiqueta Git en AWS CodeCommit

Puede utilizar una etiqueta de Git para marcar una confirmación para ayudar así a otros usuarios a comprender su importancia. Para crear una etiqueta Git en un CodeCommit repositorio, puedes usar Git desde un repositorio local conectado al CodeCommit repositorio. Una vez que hayas creado una etiqueta Git en el repositorio local, puedes usarla `git push --tags` para subirla al CodeCommit repositorio.

Para obtener más información, consulte [Visualización de información detallada de las etiquetas](#).

Uso de Git para crear una etiqueta

Sigue estos pasos para usar Git desde un repositorio local para crear una etiqueta Git en un CodeCommit repositorio.

En estos pasos, asumimos que ya has conectado el repositorio local al CodeCommit repositorio. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Ejecuta el `git tag new-tag-name commit-id` comando, donde **new-tag-name** aparece el nombre de la nueva etiqueta de Git y el ID de la confirmación que **commit-id** se va a asociar a la etiqueta de Git.

Por ejemplo, el comando siguiente crea una etiqueta de Git denominada beta y la asocia con el ID de confirmación dc082f9a...af873b88:

```
git tag beta dc082f9a...af873b88
```

2. Para enviar la nueva etiqueta Git del repositorio local al CodeCommit repositorio, ejecuta el `git push remote-name new-tag-name` comando, donde **remote-name** está el nombre del CodeCommit repositorio y **new-tag-name** es el nombre de la nueva etiqueta Git.

Por ejemplo, para enviar una nueva etiqueta de Git con el nombre beta a un CodeCommit repositorio llamado `origin`:

```
git push origin beta
```

Note

Para enviar todas las etiquetas Git nuevas de tu repositorio local al CodeCommit repositorio, ejecuta `git push --tags`.

Para asegurarte de que tu repositorio local esté actualizado con todas las etiquetas de Git del CodeCommit repositorio, ejecuta `git fetch` seguido `git fetch --tags`.

Para ver más opciones, consulte la documentación de Git.

Ver detalles de la etiqueta Git en AWS CodeCommit

En Git, una etiqueta es una marca que puede aplicarse a una referencia como una confirmación para marcarla con información que puede ser importante para otros usuarios del repositorio. Por ejemplo, puede etiquetar la confirmación que fue el punto de la versión beta para un proyecto con la etiqueta **beta**. Para obtener más información, consulte [Uso de Git para crear una etiqueta](#). Las etiquetas de Git son distintas de las etiquetas de repositorio. Para obtener más información acerca de cómo usar las etiquetas de repositorio, consulte [Añadir una etiqueta a un repositorio](#).

Puedes usar la AWS CodeCommit consola para ver información sobre las etiquetas de Git de tu repositorio, incluida la fecha y el mensaje de confirmación de la confirmación a la que hace referencia cada etiqueta de Git. Desde la consola, puede comparar la confirmación a la que hace referencia la

etiqueta con el inicio de la ramificación predeterminada del repositorio. Al igual que cualquier otra confirmación, también puede ver el código en el punto de dicha etiqueta de Git.

También puede utilizar Git desde su terminal o línea de comandos para ver detalles sobre las etiquetas de Git en un repositorio local.

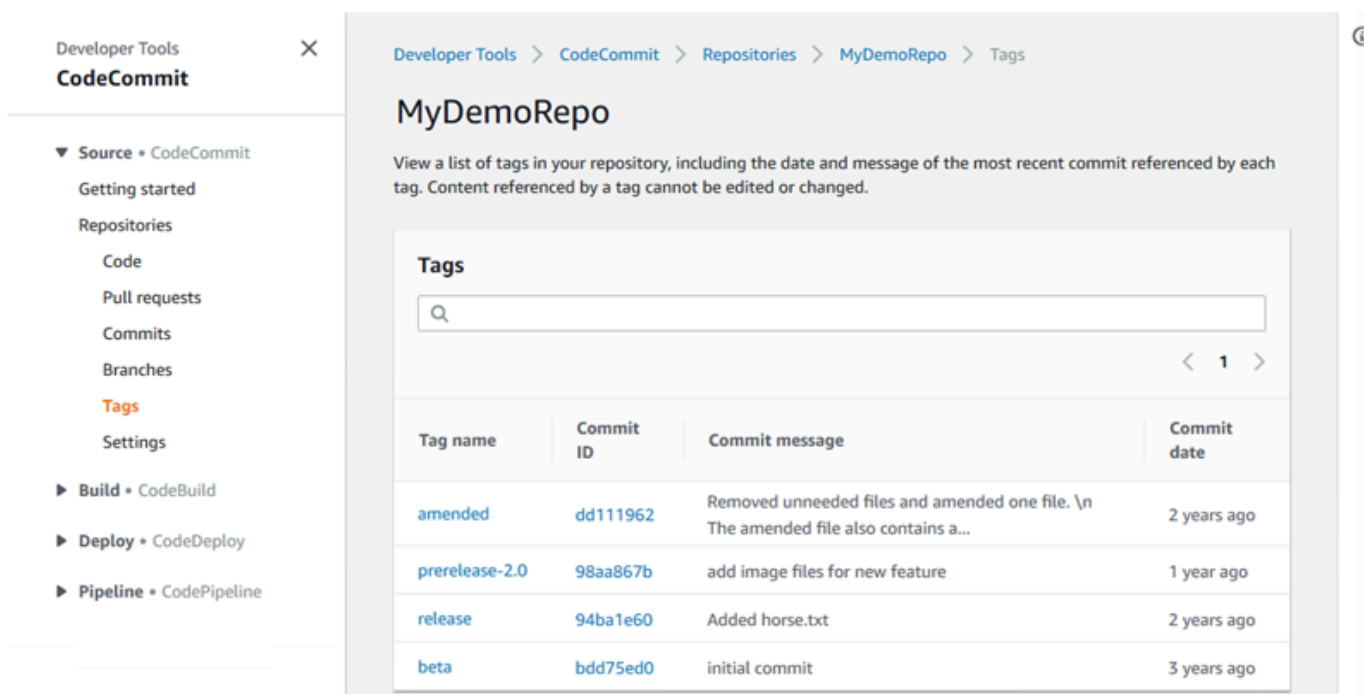
Temas

- [Ver detalles de las etiquetas \(consola\)](#)
- [Ver detalles de las etiquetas de Git \(Git\)](#)

Ver detalles de las etiquetas (consola)

Usa la AWS CodeCommit consola para ver rápidamente una lista de etiquetas de Git para tu repositorio y detalles sobre las confirmaciones a las que hacen referencia las etiquetas de Git.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas etiquetas desea ver.
3. En el panel de navegación, elija Git tags (Etiquetas de Git).



The screenshot shows the AWS CodeCommit console interface. On the left is a navigation sidebar with a 'Developer Tools' header and a 'CodeCommit' sub-header. Under 'CodeCommit', there are links for 'Getting started', 'Repositories', 'Code', 'Pull requests', 'Commits', 'Branches', 'Tags' (which is highlighted in orange), and 'Settings'. Below these are sections for 'Build' (CodeBuild), 'Deploy' (CodeDeploy), and 'Pipeline' (CodePipeline). The main content area shows the breadcrumb 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Tags'. The title 'MyDemoRepo' is displayed, followed by a description: 'View a list of tags in your repository, including the date and message of the most recent commit referenced by each tag. Content referenced by a tag cannot be edited or changed.' Below this is a 'Tags' section with a search bar and a table of tags. The table has four columns: 'Tag name', 'Commit ID', 'Commit message', and 'Commit date'. There are four tags listed: 'amended', 'prerelease-2.0', 'release', and 'beta'.

Tag name	Commit ID	Commit message	Commit date
amended	dd111962	Removed unneeded files and amended one file. \n The amended file also contains a...	2 years ago
prerelease-2.0	98aa867b	add image files for new feature	1 year ago
release	94ba1e60	Added horse.txt	2 years ago
beta	bdd75ed0	initial commit	3 years ago

4. Realice una de las siguientes acciones:
 - Para ver el código como aparecía en la confirmación, elija el nombre de la etiqueta de Git.

- Para ver los detalles de la confirmación, incluidos el mensaje de confirmación completo, el confirmador y el autor, elija el ID de confirmación abreviado.

Ver detalles de las etiquetas de Git (Git)

Para utilizar Git para ver la información detallada acerca de las etiquetas de Git en un repositorio local, ejecute uno de los siguientes comandos:

- [git tag](#) para ver una lista de los nombres de etiqueta de Git
- [git show](#) para ver información sobre una etiqueta de Git específica
- [git ls-remote](#) para ver información sobre las etiquetas de Git en un CodeCommit repositorio.

Note

Para asegurarte de que tu repositorio local esté actualizado con todas las etiquetas de Git del CodeCommit repositorio, ejecuta `git fetch` seguido `git fetch --tags` de.

En los siguientes pasos, asumimos que ya has conectado el repositorio local a un CodeCommit repositorio. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

Para ver una lista de las etiquetas de Git en un repositorio local

1. Ejecute el comando `git tag`:

```
git tag
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
beta  
release
```

Note

Si no se han definido etiquetas, `git tag` no dará ningún resultado.

Para ver más opciones, consulte la documentación de Git.

Para ver información sobre una etiqueta de Git en un repositorio local

1. Ejecute el comando `git show tag-name`. Por ejemplo, para ver información sobre una etiqueta de Git denominada `beta`, ejecute:

```
git show beta
```

2. Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
commit 317f8570...ad9e3c09
Author: John Doe <johndoe@example.com>
Date:   Tue Sep 23 13:49:51 2014 -0700

    Added horse.txt

diff --git a/horse.txt b/horse.txt
new file mode 100644
index 0000000..df42ff1
--- /dev/null
+++ b/horse.txt
@@ -0,0 +1 @@
+The horse (Equus ferus caballus) is one of two extant subspecies of Equus ferus
\ No newline at end of file
```

Note

Para salir del resultado de la información de la etiqueta de Git, introduzca `:q`.

Para ver más opciones, consulte la documentación de Git.

Para ver información sobre las etiquetas de Git en un CodeCommit repositorio

1. Ejecute el comando `git ls-remote --tags`.

```
git ls-remote --tags
```

2. Si se ejecuta correctamente, este comando genera como resultado una lista de las etiquetas Git del CodeCommit repositorio:

```
129ce87a...70fbffba    refs/tags/beta
785de9bd...59b402d8    refs/tags/release
```

Si no se han definido etiquetas de Git, `git ls-remote --tags` devolverá una línea en blanco.

Para ver más opciones, consulte la documentación de Git.

Eliminar una etiqueta de Git en AWS CodeCommit

Para eliminar una etiqueta de Git en un CodeCommit repositorio, usa Git de un repositorio local conectado al CodeCommit repositorio.

Uso de Git para eliminar una etiqueta de Git

Sigue estos pasos para usar Git desde un repositorio local para eliminar una etiqueta de Git en un CodeCommit repositorio.

Estos pasos se escriben partiendo del supuesto de que ya has conectado el repositorio local al CodeCommit repositorio. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Para eliminar la etiqueta Git del repositorio local, ejecuta el `git tag -d tag-name` comando donde **tag-name** está el nombre de la etiqueta Git que deseas eliminar.

Tip

Para obtener una lista de nombres de etiqueta de Git, ejecute `git tag`.

Por ejemplo, para eliminar una etiqueta de Git del repositorio local denominado `beta`:

```
git tag -d beta
```

2. Para eliminar la etiqueta Git del CodeCommit repositorio, ejecuta el `git push remote-name --delete tag-name` comando donde **remote-name** es el apodo que el repositorio local usa para el CodeCommit repositorio y **tag-name** es el nombre de la etiqueta Git que deseas eliminar del CodeCommit repositorio.

 Tip

Para obtener una lista de los nombres de los CodeCommit repositorios y sus direcciones URL, ejecuta el `git remote -v` comando.

Por ejemplo, para eliminar una etiqueta de Git denominada `beta` en el CodeCommit repositorio `origin`:

```
git push origin --delete beta
```

Trabajando con ramas en AWS CodeCommit repositorios

¿Qué es una ramificación? 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. Puede utilizar las ramificaciones para desarrollar nuevas características, guardar una versión específica del proyecto de una determinada confirmación y mucho más. Al crear su primera confirmación, se crea una ramificación predeterminada para usted. Esta ramificación predeterminada es la que se utiliza como base o ramificación predeterminada en los repositorios (repos) locales cuando los usuarios clonan el repositorio. El nombre de esa ramificación predeterminada varía en función de cómo cree su primera confirmación. Si agregas el primer archivo al repositorio mediante la CodeCommit consola AWS CLI, el SDK o uno de los SDK, el nombre de esa rama predeterminada es main. Esta es la ramificación predeterminada en todos los ejemplos de esta guía. Si envía su primera confirmación con un cliente Git, el nombre de la ramificación predeterminada es el que el cliente Git especifique como predeterminado. Considere configurar su cliente de Git para que use principal como nombre de la ramificación inicial.

En CodeCommit, puedes cambiar la rama predeterminada de tu repositorio. También puede crear y eliminar ramificaciones y consultar información más detallada sobre una ramificación. Puede comparar rápidamente una ramificación con la ramificación predeterminada (o dos ramificaciones cualesquiera). Para ver el historial de ramificaciones y combinaciones del repositorio, puede utilizar [Commit Visualizer](#), que se muestra en el siguiente gráfico.

The screenshot displays the AWS CodeCommit web interface for a repository named 'MyDemoRepo'. The 'Commit visualizer' tab is active, showing a graphical representation of the commit history on the left and a list of commits on the right. The commits are ordered from most recent at the top to older ones at the bottom.

Commit Hash	Commit Message	Time Ago
d615e7ae	Merge branch 'AnotherBranch' into testbranch	2 minutes ago
b6589863	Added another file.	2 minutes ago
73a6e39c	remote-tracking branch 'refs/remotes/origin/jane-branch' into jane-branch	
6bbb6d3c	Another test of the editing feature.	20 minutes ago
edacdfef	Testing this out to see how well it works.	23 minutes ago
70bb94d7	Revised test results with correct information.	36 minutes ago
b78e6d1c	Merge branch 'results' into testbranch	50 minutes ago
84b7d158	Edited ahs_count.py	50 minutes ago

Para obtener información sobre cómo trabajar con otros aspectos de tu repositorio en CodeCommit [Trabajar con repositorios](#), [Trabajar con archivos](#), [Trabajo con solicitudes de extracción](#), [Uso de confirmaciones](#), y [Trabajo con las preferencias del usuario](#).

Temas

- [Crea una rama en AWS CodeCommit](#)
- [Limite las inserciones y fusiones a las ramificaciones AWS CodeCommit](#)
- [Consulta los detalles de la sucursal en AWS CodeCommit](#)
- [Compara y fusiona ramas en AWS CodeCommit](#)
- [Cambie la configuración de la sucursal en AWS CodeCommit](#)
- [Eliminar una rama en AWS CodeCommit](#)

Crea una rama en AWS CodeCommit

Puede utilizar la CodeCommit consola o la AWS CLI para crear ramas para el repositorio. Se trata de una forma rápida de separar el trabajo en una versión de los archivos nueva o diferente sin que ello afecte al trabajo en la ramificación predeterminada. Después de crear una rama en la CodeCommit consola, debes transferir ese cambio a tu repositorio local. Como alternativa, puedes crear una rama localmente y luego usar Git desde un repositorio local conectado al CodeCommit repositorio para impulsar ese cambio.

Temas

- [Creación de una ramificación \(consola\)](#)
- [Creación de una ramificación \(Git\)](#)
- [Crea una rama \(AWS CLI\)](#)

Creación de una ramificación (consola)

Puedes usar la CodeCommit consola para crear una rama en un CodeCommit repositorio. La próxima vez que los usuarios envíen cambios desde el repositorio, ven la nueva ramificación.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio en el que desea crear la ramificación.
3. En el panel de navegación, seleccione Branches.
4. Seleccione Create branch.

Create branch

Branch name
feature_advanced-class

Branch from

Q |Type to filter.

Branches

- main
Default branch
- bugfix-1236
- feature-lambdafunctions
- feature-new-wizard
- feature-randomizationfeature

Git tags

- release
- prerelease-2.0
- beta
- amended

Create branch

En Branch name (Nombre de ramificación), introduzca un nombre para la ramificación. En Branch from (Origen de ramificación), seleccione una ramificación o una etiqueta de la lista o pegue un ID de confirmación. Seleccione Create branch.

Creación de una ramificación (Git)

Sigue estos pasos para usar Git desde un repositorio local para crear una rama en un repositorio local y luego enviar esa rama al CodeCommit repositorio.

Estos pasos se escriben partiendo del supuesto de que ya has conectado el repositorio local al repositorio. CodeCommit Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Crea una rama en tu repositorio local ejecutando el git checkout -b **new-branch-name** comando, donde **new-branch-name** está el nombre de la nueva rama.

Por ejemplo, el siguiente comando crea una ramificación denominada MyNewBranch en el repositorio local:

```
git checkout -b MyNewBranch
```

2. Para enviar la nueva rama del repositorio local al CodeCommit repositorio, ejecuta el git push comando especificando tanto la rama como **remote-name** la. **new-branch-name**

Por ejemplo, para insertar una nueva rama en el repositorio local con el nombre MyNewBranch del CodeCommit repositorio con el apodo: origin

```
git push origin MyNewBranch
```

Note

Si agrega la -u opción a git push (por ejemplo, git push -u origin main), en el futuro podrá ejecutar git push sin ella **remote-namebranch-name**. Se establece la información de seguimiento ascendente. Para obtener información de seguimiento ascendente, ejecute git remote show **remote-name** (por ejemplo, git remote show origin).

Para ver una lista de todas las ramificaciones de seguimiento remoto, ejecute git branch --all. Para configurar una rama en el repositorio local que esté conectada a una rama del CodeCommit repositorio, ejecute git checkout **remote-branch-name**.

Para ver más opciones, consulte la documentación de Git.

Crea una rama (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Siga estos pasos para usar el AWS CLI para crear una rama en un CodeCommit repositorio y, a continuación, enviar esa rama al CodeCommit repositorio. Para ver los pasos para crear una confirmación inicial y especificar el nombre de la ramificación predeterminada para un repositorio vacío, consulte [Crear la primera confirmación para un repositorio mediante AWS CLI](#).

1. Ejecute el comando `create-branch`, especificando:

- El nombre del CodeCommit repositorio en el que se crea la rama (con la `--repository-name` opción).

 Note

Para obtener el nombre del CodeCommit repositorio, ejecute el comando [list-repositories](#).

- El nombre de la nueva ramificación (con la opción `--branch-name`).
- El ID de la confirmación a la que apunta la nueva ramificación (con la opción `--commit-id`).

Por ejemplo, para crear una rama con un nombre `MyNewBranch` que apunte al ID de confirmación `317f8570EXAMPLE` en un CodeCommit repositorio llamado: `MyDemoRepo`

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

Este comando devuelve resultados solo si hay errores.

2. Para actualizar la lista de ramas de CodeCommit repositorio disponibles en tu repositorio local con el nombre de la nueva rama remota, ejecute `git remote update remote-name`.

Por ejemplo, para actualizar la lista de sucursales disponibles para el CodeCommit repositorio con el apodo `origin`:

```
git remote update origin
```

 Note

Si lo desea, también puede ejecutar el comando `git fetch`. También puede ver todas las ramificaciones remotas ejecutando `git branch --all`, pero hasta que actualice la lista de su repositorio local, la ramificación remota que ha creado no aparece en la lista. Para ver más opciones, consulte la documentación de Git.

3. Para configurar una rama en el repositorio local que esté conectada a la nueva rama del CodeCommit repositorio, ejecute `git checkout remote-branch-name`.

 Note

Para obtener una lista de los nombres de los CodeCommit repositorios y sus direcciones URL, ejecute el `git remote -v` comando.

Limite las inserciones y fusiones a las ramificaciones AWS CodeCommit

De forma predeterminada, cualquier usuario del CodeCommit repositorio que tenga permisos suficientes para insertar código en el repositorio puede contribuir a cualquier rama de ese repositorio. Esto se aplica independientemente de cómo se agregue una ramificación al repositorio, ya sea desde la consola, la línea de comandos o Git. Sin embargo, es posible que desee configurar una ramificación de tal forma que solo algunos usuarios del repositorio puedan insertar o combinar el código en ella. Por ejemplo, puede que desee configurar una ramificación que se use para el código de producción de modo que únicamente un subconjunto de los desarrolladores sénior puedan insertar o combinar cambios en ella. Los demás desarrolladores podrán extraer código de la ramificación, crear sus propias ramificaciones y crear solicitudes de extracción, pero no podrán insertar ni combinar cambios en ella. Para configurar este acceso, se crea una política condicional que usa una clave de contexto para una o varias ramificaciones de IAM.

 Note

Para completar algunos de los procedimientos de este tema, debe iniciar sesión con un usuario administrativo que tenga permisos suficientes para configurar y aplicar políticas de

IAM. Para obtener más información, consulte [Creación de un grupo y usuario administrador de IAM](#).

Temas

- [Configuración de una política de IAM para limitar las inserciones y combinaciones en una ramificación](#)
- [Aplicación de una política de IAM a un grupo o rol de IAM](#)
- [Comprobación de la política](#)

Configuración de una política de IAM para limitar las inserciones y combinaciones en una ramificación

Puede crear una política en IAM que impida que los usuarios actualicen una ramificación, lo que incluye insertar confirmaciones y combinar solicitudes de extracción en ella. Para ello, la política usa una instrucción condicional, de tal forma que el efecto de la instrucción Deny se aplique únicamente si se cumple la condición. Las API que se incluyen en la instrucción Deny determinan qué acciones no se permiten. Puede configurar esta política de modo que solo se aplique a una ramificación del repositorio, a varias de ellas en el mismo repositorio o a todas las ramificaciones que coincidan con los criterios de todos los repositorios de una cuenta de Amazon Web Services.

Para crear una política condicional para las ramificaciones

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, seleccione Políticas.
3. Elija Crear política.
4. Elija JSON y, a continuación, pegue la siguiente política de ejemplo. Reemplace el valor de Resource por el ARN del repositorio que contenga la ramificación a la que desee restringir el acceso. Reemplace el valor `codecommit:References` por una referencia a la ramificación o ramificaciones a las que desee restringir el acceso. Por ejemplo, esta política deniega el envío de confirmaciones, la fusión de ramificaciones, la eliminación de ramificaciones, la eliminación de archivos, la fusión de solicitudes de extracción y la adición de archivos a una ramificación denominada *main* y a una ramificación denominada *prod* en un repositorio denominado *MyDemoRepo*:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecommit:GitPush",
        "codecommit>DeleteBranch",
        "codecommit>DeleteFile",
        "codecommit:PutFile",
        "codecommit:MergeBranchesByFastForward",
        "codecommit:MergeBranchesBySquash",
        "codecommit:MergeBranchesByThreeWay",
        "codecommit:MergePullRequestByFastForward",
        "codecommit:MergePullRequestBySquash",
        "codecommit:MergePullRequestByThreeWay"
      ],
      "Resource": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
      "Condition": {
        "StringEqualsIfExists": {
          "codecommit:References": [
            "refs/heads/main",
            "refs/heads/prod"
          ]
        },
        "Null": {
          "codecommit:References": "false"
        }
      }
    }
  ]
}
```

Las ramas en Git son simplemente indicadores (referencias) al SHA-1 valor de la confirmación principal, razón por la cual se usa References la condición. La instrucción Null es obligatoria en todas las políticas cuyo efecto sea Deny y en las que una de las acciones sea GitPush. Es

obligatoria por la forma en que funcionan Git y `git-receive-pack` al insertar los cambios de un repositorio local en CodeCommit.

 Tip

Para crear una política que se aplique a todas las ramificaciones denominadas "principal" de todos los repositorios de una cuenta de Amazon Web Services, cambie el valor de `Resource` por un asterisco (*) en lugar del ARN del repositorio.

5. Elija Revisar política. Corrija los errores de la instrucción de la política y, a continuación, continúe con Create policy (Crear política).
6. Cuando se valida el código JSON, aparece la página Create policy (Crear política). Se muestra una advertencia en la sección Summary (Resumen) que le informa de que esta política no concede permisos. Esto es normal.
 - En Name (Nombre), escriba un nombre para esta política; por ejemplo, **DenyChangesToMain**.
 - En Description (Descripción), introduzca una descripción de la finalidad de la política. Esto es opcional, pero recomendable.
 - Elija Crear política.

Aplicación de una política de IAM a un grupo o rol de IAM

Ha creado una política que limita las inserciones y combinaciones en una ramificación. Sin embargo, la política no surte ningún efecto hasta que se aplica a un usuario, grupo o rol de IAM. La práctica recomendada es aplicar la política a un grupo o función de IAM. Si las políticas se aplican a usuarios de IAM individuales, el escalado no se realiza correctamente.

Para aplicar la política condicional a un grupo o función

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
2. En el panel de navegación, si desea aplicar la política a un grupo de IAM, seleccione Grupos. Si desea aplicar la política a una función asumida por los usuarios, seleccione Rol. Seleccione el nombre del grupo o la función.
3. En la pestaña Permissions, elija Attach Policy.

4. Seleccione la política condicional que ha creado en la lista de políticas y, a continuación, elija **Attach policy** (Adjuntar política).

Para obtener más información, consulte [Conectar y separar políticas de IAM](#).

Comprobación de la política

Debe probar los efectos de la política que ha aplicado al grupo o función para asegurarse de que actúa según lo previsto. Existen muchas formas de realizar esta operación. Por ejemplo, para probar una política similar a la que se muestra anteriormente, puede hacer lo siguiente:

- Inicie sesión en la CodeCommit consola con un usuario de IAM que sea miembro de un grupo de IAM al que se haya aplicado la política o que asuma un rol al que se le haya aplicado la política. En la consola, agregue un archivo en la ramificación a la que se aplican las restricciones. Debería aparecer un mensaje de error al intentar guardar o cargar un archivo en esa ramificación. Agregue un archivo a otra ramificación. La operación debería realizarse correctamente.
- Inicie sesión en la CodeCommit consola con un usuario de IAM que sea miembro de un grupo de IAM al que se haya aplicado la política o que asuma un rol al que se le haya aplicado la política. Cree una solicitud de extracción que se combine con la ramificación a la que se aplican las restricciones. Debería poder crear la solicitud de extracción, pero debería aparecer un error si intenta combinarla.
- Desde el terminal o la línea de comandos, crea una confirmación en la rama en la que se aplican las restricciones y, a continuación, envía esa confirmación al CodeCommit repositorio. Debería aparecer un mensaje de error. Las confirmaciones e inserciones realizadas desde otras ramificaciones deberían funcionar normalmente.

Consulta los detalles de la sucursal en AWS CodeCommit

Puede usar la CodeCommit consola para ver detalles sobre las ramas de un CodeCommit repositorio. Puede ver la fecha de la última conformación de una ramificación, el mensaje de confirmación y mucho más. También puedes usar Git AWS CLI o Git desde un repositorio local conectado al CodeCommit repositorio.

Temas

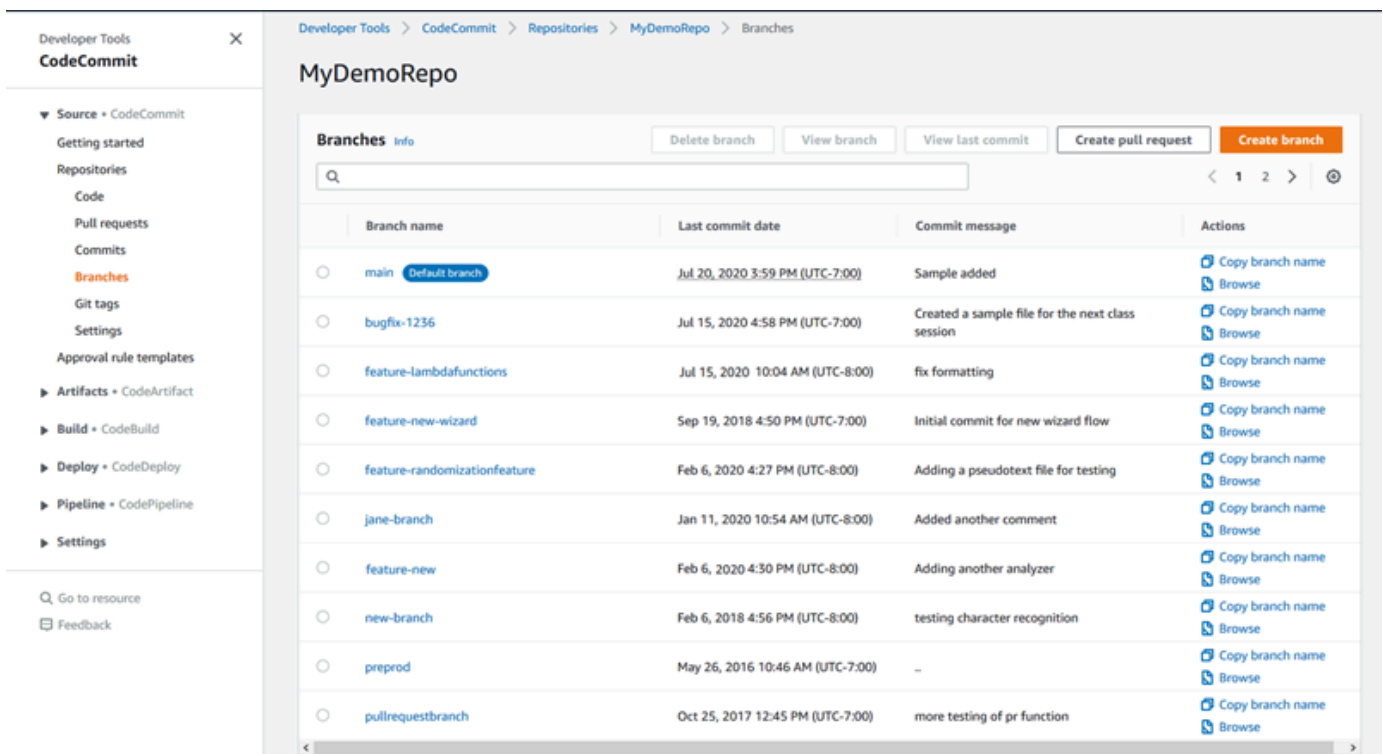
- [Ver detalles de las ramificaciones \(consola\)](#)
- [Visualización de la información detallada de la ramificación \(Git\)](#)

- [Ver detalles de la sucursal \(AWS CLI\)](#)

Ver detalles de las ramificaciones (consola)

Usa la CodeCommit consola para ver rápidamente una lista de las ramas de tu repositorio y los detalles sobre las ramas.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyos detalles de ramificación desea ver.
3. En el panel de navegación, seleccione Branches.



The screenshot shows the AWS CodeCommit console interface. On the left is a navigation pane with options like 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main area displays the 'MyDemoRepo' repository details. At the top, there are buttons for 'Delete branch', 'View branch', 'View last commit', 'Create pull request', and 'Create branch'. Below this is a table of branches.

Branch name	Last commit date	Commit message	Actions
main (Default branch)	Jul 20, 2020 3:59 PM (UTC-7:00)	Sample added	Copy branch name, Browse
bugfix-1236	Jul 15, 2020 4:58 PM (UTC-7:00)	Created a sample file for the next class session	Copy branch name, Browse
feature-lambdafunctions	Jul 15, 2020 10:04 AM (UTC-8:00)	fix formatting	Copy branch name, Browse
feature-new-wizard	Sep 19, 2018 4:50 PM (UTC-7:00)	Initial commit for new wizard flow	Copy branch name, Browse
feature-randomizationfeature	Feb 6, 2020 4:27 PM (UTC-8:00)	Adding a pseudotext file for testing	Copy branch name, Browse
jane-branch	Jan 11, 2020 10:54 AM (UTC-8:00)	Added another comment	Copy branch name, Browse
feature-new	Feb 6, 2020 4:30 PM (UTC-8:00)	Adding another analyzer	Copy branch name, Browse
new-branch	Feb 6, 2018 4:56 PM (UTC-8:00)	testing character recognition	Copy branch name, Browse
preprod	May 26, 2016 10:46 AM (UTC-7:00)	--	Copy branch name, Browse
pullrequestbranch	Oct 25, 2017 12:45 PM (UTC-7:00)	more testing of pr function	Copy branch name, Browse

4. El nombre de la ramificación que se utiliza como nombre predeterminado para el repositorio aparece junto a la etiqueta Ramificación predeterminada. Para ver información detallada sobre las últimas confirmaciones en una ramificación, elija la ramificación y, a continuación, elija View last commit (Ver última confirmación). Para ver los archivos y el código en una ramificación, elija el nombre de la ramificación.

Visualización de la información detallada de la ramificación (Git)

Para usar Git desde un repositorio local para ver detalles sobre las ramas de seguimiento local y remoto de un CodeCommit repositorio, ejecuta el `git branch` comando.

Los siguientes pasos se escriben partiendo del supuesto de que ya has conectado el repositorio local al CodeCommit repositorio. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Ejecute el comando `git branch` y especifique la opción `--all`:

```
git branch --all
```

2. Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
MyNewBranch
* main
remotes/origin/MyNewBranch
remotes/origin/main
```

El asterisco (*) aparece junto a la ramificación abierta actualmente. Las entradas que aparecen después son referencias de seguimiento remoto.

Tip

`git branch` muestra las ramificaciones locales.
`git branch -r` muestra las ramificaciones remotas.
`git checkout existing-branch-name` cambia al nombre de ramificación especificado y, si `git branch` se ejecuta inmediatamente después, lo muestra con un asterisco (*).
`git remote update remote-name` actualiza tu repositorio local con la lista de ramas de CodeCommit repositorio disponibles. (Para obtener una lista de los nombres de los CodeCommit repositorios y sus URL, ejecuta el `git remote -v` comando).

Para ver más opciones, consulte la documentación de Git.

Ver detalles de la sucursal (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar el AWS CLI para ver detalles sobre las ramas de un CodeCommit repositorio, ejecute uno o más de los siguientes comandos:

- Ejecute [list-branches](#) para ver una lista de los nombres de ramificaciones.
- Ejecute [get-branch](#) para ver información sobre una ramificación específica.

Visualización de una lista de los nombres de ramificaciones

1. Ejecute el list-branches comando y especifique el nombre del CodeCommit repositorio (con la --repository-name opción).

Tip

Para obtener el nombre del CodeCommit repositorio, ejecute el comando [list-repositories](#).

Por ejemplo, para ver los detalles de las ramas de un CodeCommit repositorio llamado: MyDemoRepo

```
aws codecommit list-branches --repository-name MyDemoRepo
```

2. Si se ejecuta correctamente, este comando emite un objeto branchNameList con una entrada para cada ramificación.

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "branches": [
    "MyNewBranch",
    "main"
  ]
}
```

Para ver información sobre una ramificación

1. Ejecute el comando get-branch, especificando:

- El nombre del repositorio (con la opción `--repository-name`).
- El nombre de la ramificación (con la opción `--branch-name`).

Por ejemplo, para ver información sobre una rama denominada `MyNewBranch` en un CodeCommit repositorio denominado `MyDemoRepo`:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

2. Si se ejecuta correctamente, este comando emite el nombre de la ramificación y el ID de la última confirmación realizada a la ramificación.

A continuación se muestra un ejemplo de resultado basado en el comando del ejemplo anterior:

```
{
  "branch": {
    "branchName": "MyNewBranch",
    "commitID": "317f8570EXAMPLE"
  }
}
```

Compara y fusiona ramas en AWS CodeCommit

Puede usar la CodeCommit consola para comparar las ramas de un CodeCommit repositorio. La comparación de ramificaciones le permite ver rápidamente las diferencias que existen entre una ramificación y la ramificación predeterminada, o ver las diferencias que existen entre dos ramificaciones.

Temas

- [Comparación de una ramificación y la ramificación predeterminada](#)
- [Comparación de dos ramificaciones específicas](#)
- [Combina dos ramas \(AWS CLI\)](#)

Comparación de una ramificación y la ramificación predeterminada

Utilice la CodeCommit consola para ver rápidamente las diferencias entre una rama y la rama predeterminada de su repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas ramificaciones desea comparar.
3. En el panel de navegación, seleccione Commits (Confirmaciones) y haga clic en la pestaña Compare commits (Comparar confirmaciones).
4. En Destination (Destino), seleccione el nombre de la ramificación predeterminada. En Source (Origen), elija la ramificación que desee comparar con la ramificación predeterminada. Elija Compare.

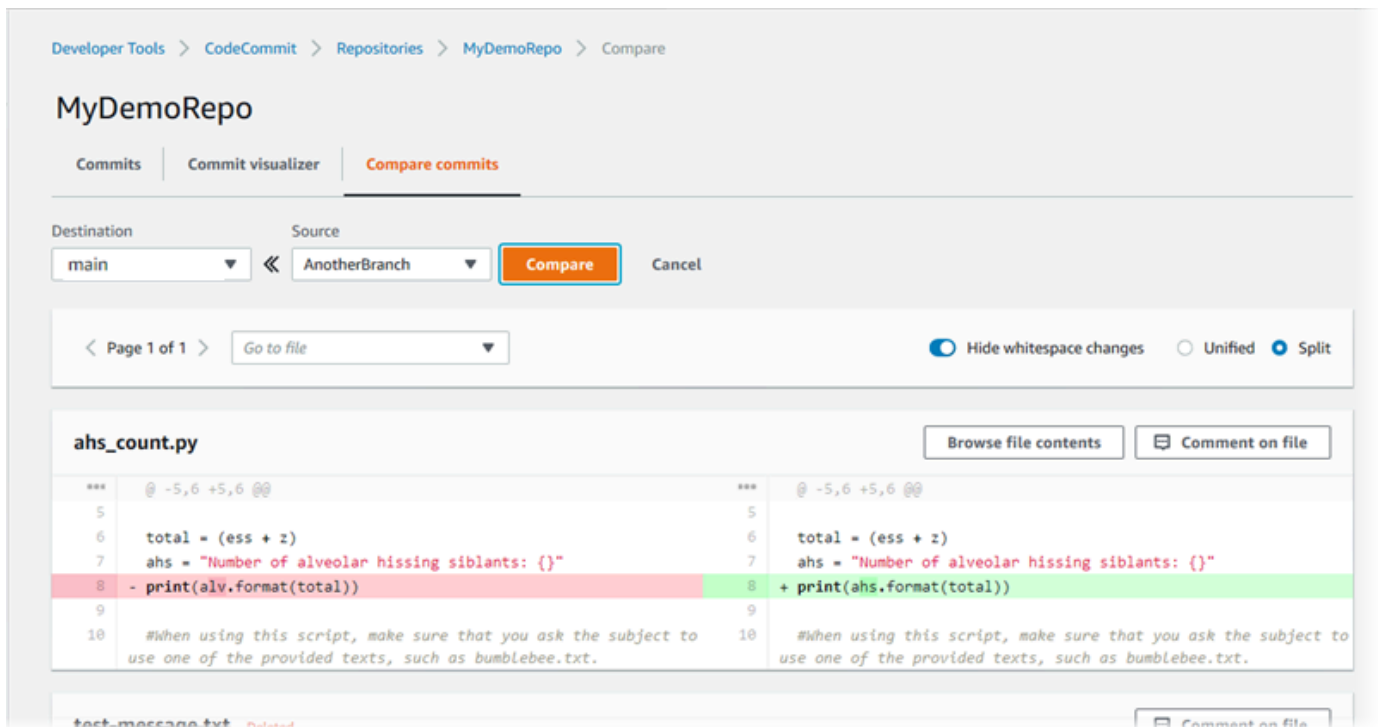
Comparación de dos ramificaciones específicas

Utilice la CodeCommit consola para ver las diferencias entre las dos ramas que desee comparar.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuyas ramificaciones desea comparar.
3. En el panel de navegación, seleccione Commits (Confirmaciones) y haga clic en la pestaña Compare commits (Comparar confirmaciones).
4. En Destination (Destino) y Source (Origen), seleccione las dos ramificaciones que desea comparar y haga clic en Compare (Comparar). Para ver la lista de archivos modificados, amplíe la lista de archivos modificados. Puede ver los cambios en los archivos de forma paralela (vista Split) o en línea (vista Unified).

Note

Si ha iniciado sesión como un usuario de IAM, puede configurar y guardar sus preferencias para ver el código y otros ajustes de la consola. Para obtener más información, consulte [Trabajo con las preferencias del usuario](#).



Combina dos ramas (AWS CLI)

Puede combinar dos ramas en un CodeCommit repositorio AWS CLI utilizando una de las estrategias de combinación disponibles ejecutando uno de los siguientes comandos:

- Para combinar dos ramificaciones mediante la estrategia de combinación de avance rápido, ejecute el comando [merge-branches-by-fast-forward](#).
- Para combinar dos ramificaciones mediante la estrategia de combinación de squash, ejecute el comando [merge-branches-by-squash](#).
- Para combinar dos ramificaciones mediante la estrategia de combinación de tres modos, ejecute el comando [merge-branches-by-three-way](#).

También puede probar las combinaciones ejecutando el comando `create-unreferenced-merge-commit`. Para obtener más información, consulte [Resolver conflictos en una solicitud de extracción](#).

 Note

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para utilizar el AWS CLI para fusionar dos ramas en un CodeCommit repositorio

1. Para combinar dos ramificaciones mediante la estrategia de combinación de avance rápido, ejecute el comando `merge-branches-by-fast-forward` especificando:
 - El nombre de la ramificación de origen que contiene los cambios que desea combinar (con la opción `--source-commit-specifier`).
 - El nombre de la ramificación de destino donde desea combinar los cambios (con la opción `--destination-commit-specifier`).
 - El nombre del repositorio (con la opción `--repository-name`).

Por ejemplo, para fusionar una rama de origen denominada *bugfix-1234* con una rama de destino denominada *preprod* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit merge-branches-by-fast-forward --source-commit-specifier bugfix-1234 --destination-commit-specifier preprod --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

2. Para combinar dos ramificaciones mediante la estrategia de combinación de squash, ejecute el comando `merge-branches-by-squash`, especificando:
 - El nombre de la ramificación de origen que contiene los cambios que desea combinar (con la opción `--source-commit-specifier`).
 - El nombre de la ramificación de destino donde desea combinar los cambios (con la opción `--destination-commit-specifier`).

- El nombre del repositorio (con la opción `--repository-name`).
- El mensaje de confirmación que incluir (con la opción `--commit-message`).
- El nombre que se utilizará para la confirmación (con la opción `--name`).
- La dirección de correo electrónico que se va a utilizar para la confirmación (con la opción `--email`).

Por ejemplo, para fusionar una rama de origen denominada *bugfix-bug1234* con una rama de destino denominada *bugfix-quarterly* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit merge-branches-by-squash --source-commit-specifier bugfix-bug1234 --  
destination-commit-specifier bugfix-quarterly --author-name "Maria Garcia" --email  
"maria_garcia@example.com" --commit-message "Merging in fix branches to prepare  
for a general patch." --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

3.

Para combinar dos ramificaciones mediante la estrategia de combinación de tres modos, ejecute el comando `merge-branches-by-three-way`, especificando:

- El nombre de la ramificación de origen que contiene los cambios que desea combinar (con la opción `--source-commit-specifier`).
- El nombre de la ramificación de destino donde desea combinar los cambios (con la opción `--destination-commit-specifier`).
- El nombre del repositorio (con la opción `--repository-name`).
- El mensaje de confirmación que incluir (con la opción `--commit-message`).
- El nombre que se utilizará para la confirmación (con la opción `--name`).
- La dirección de correo electrónico que se va a utilizar para la confirmación (con la opción `--email`).

Por ejemplo, para fusionar una rama de origen denominada *main* con una rama de destino denominada *bugfix-1234* en un repositorio denominado *MyDemoRepo*:

```
aws codecommit merge-branches-by-three-way --source-commit-specifier main --  
destination-commit-specifier bugfix-bug1234 --author-name "Jorge Souza" --email  
"jorge_souza@example.com" --commit-message "Merging changes from main to bugfix  
branch before additional testing." --repository-name MyDemoRepo
```

Si se ejecuta correctamente, el resultado del comando será similar al siguiente:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Cambie la configuración de la sucursal en AWS CodeCommit

Puede cambiar la rama que desea utilizar como rama predeterminada en la AWS CodeCommit consola o con la AWS CLI. Por ejemplo, si creó su primera confirmación con un cliente Git que estableció la ramificación predeterminada como maestro, podría crear una ramificación llamada principal y, a continuación, cambiar la configuración para que la nueva ramificación se establezca como la predeterminada del repositorio. Para cambiar la configuración de otras ramas, puedes usar Git desde un repositorio local conectado al CodeCommit repositorio.

Temas

- [Cambiar la ramificación predeterminada \(consola\)](#)
- [Cambie la rama predeterminada \(AWS CLI\)](#)

Cambiar la ramificación predeterminada (consola)

Puedes especificar qué rama es la rama predeterminada de un CodeCommit repositorio en la AWS CodeCommit consola.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio cuya configuración desea cambiar.

3. En el panel de navegación, seleccione Configuración.
4. En la lista desplegable Default branch (Ramificación predeterminada), elija la lista desplegable de ramificación y elija una ramificación diferente. Seleccione Save.

 Tip

- Si no ve otra ramificación en la lista desplegable, significa que no ha creado ninguna ramificación adicional. No puede cambiar la ramificación predeterminada de un repositorio si el repositorio solo tiene una ramificación. Para obtener más información, consulte [Crea una rama en AWS CodeCommit](#).
- Si no ve la sección Ramificación predeterminada, sino elementos relacionados con las reglas de notificación y las conexiones, se encuentra en el menú de configuración general de la consola. El menú de configuración de los repositorios aparece en Repositorios, al mismo nivel que Código y Solicitudes de extracción.

Cambie la rama predeterminada (AWS CLI)

Para usar AWS CLI comandos con CodeCommit, instale el AWS CLI. Para obtener más información, consulte [Referencia de la línea de comandos](#).

Para usar el AWS CLI para cambiar la configuración de la rama de un repositorio en un CodeCommit repositorio, ejecuta el siguiente comando:

- [update-default-branch](#) para cambiar la ramificación predeterminada.

Cambiar la ramificación predeterminada

1. Ejecute el comando update-default-branch, especificando:
 - El nombre del CodeCommit repositorio en el que se actualiza la rama predeterminada (con la --repository-name opción).

 Tip

Para obtener el nombre del CodeCommit repositorio, ejecute el comando [list-repositories](#).

- El nombre de la nueva ramificación predeterminada (con la opción `--default-branch-name`).

 Tip

Para obtener el nombre de la ramificación, ejecute el comando [list-branches](#).

2. Por ejemplo, para cambiar la rama predeterminada a la de un MyNewBranch CodeCommit repositorio llamado: MyDemoRepo

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Este comando devuelve resultados solo si hay errores.

Para ver más opciones, consulte la documentación de Git.

Eliminar una rama en AWS CodeCommit

Puedes usar la CodeCommit consola para eliminar una rama de un repositorio. Al eliminar una rama de un repositorio local, CodeCommit no se elimina esa rama de un repositorio local, por lo que es posible que los usuarios sigan teniendo copias de esa rama hasta la próxima vez que consulten los cambios. Para eliminar una rama localmente y enviar ese cambio al CodeCommit repositorio, usa Git desde un repositorio local conectado al CodeCommit repositorio.

Al eliminar una ramificación, no se elimina ninguna confirmación, pero sí todas las referencias a las confirmaciones en esa ramificación. Si elimina una ramificación que contiene confirmaciones que no se han combinado en otra ramificación en el repositorio, no puede recuperar esas confirmaciones a menos que tenga sus ID de confirmación completos.

 Note

No puede usar las instrucciones de este tema para eliminar la ramificación predeterminada de un repositorio. Si desea eliminar la ramificación predeterminada, deberá crear una ramificación, establecer esta como predeterminada y, a continuación, eliminar la antigua. Para obtener más información, consulte [Creación de una ramificación](#) y [Cambiar la configuración de una ramificación](#).

Temas

- [Eliminación de una ramificación \(consola\)](#)
- [Eliminar una rama \(AWS CLI\)](#)
- [Eliminación de una ramificación \(Git\)](#)

Eliminación de una ramificación (consola)

Puedes usar la CodeCommit consola para eliminar una rama de un CodeCommit repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositories (Repositorios), elija el nombre del repositorio en el que desea eliminar la ramificación.
3. En el panel de navegación, seleccione Branches.
4. Encuentre el nombre de la ramificación que desee eliminar, seleccione Delete branch (Eliminar ramificación) y confirme su elección.

Eliminar una rama (AWS CLI)

Puede usar la AWS CLI para eliminar una rama de un CodeCommit repositorio, si esa rama no es la rama predeterminada del repositorio. Para obtener más información sobre la instalación y el uso de AWS CLI, consulte [Referencia de la línea de comandos](#).

1. En el terminal o en la línea de comandos, ejecute el comando delete-branch especificando lo siguiente:
 - El nombre del CodeCommit repositorio en el que se va a eliminar la rama (con la --repository-name opción).

Tip

Para obtener el nombre del CodeCommit repositorio, ejecute el comando [list-repositories](#).

- El nombre de la nueva ramificación que se va a eliminar (con la opción branch-name).

 Tip

Para obtener el nombre de la ramificación, ejecute el comando [list-branches](#).

2. Por ejemplo, para eliminar una rama denominada MyNewBranch en un CodeCommit repositorio con el nombre: MyDemoRepo

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Este comando devuelve información acerca de la ramificación eliminada, incluido el nombre de la ramificación eliminada y el ID de confirmación completo que era el encabezado de la ramificación. Por ejemplo:

```
"deletedBranch": {  
  "branchName": "MyNewBranch",  
  "commitId": "317f8570EXAMPLE"  
}
```

Eliminación de una ramificación (Git)

Sigue estos pasos para usar Git desde un repositorio local para eliminar una rama de un CodeCommit repositorio.

Estos pasos se escriben partiendo del supuesto de que ya has conectado el repositorio local al CodeCommit repositorio. Para obtener instrucciones, consulte [Conectarse a un repositorio](#).

1. Para eliminar la rama del repositorio local, ejecuta el git branch -D **branch-name** comando donde **branch-name** aparece el nombre de la rama que deseas eliminar.

 Tip

Para obtener una lista de los nombres de las ramificaciones, ejecute git branch --all.

Por ejemplo, para eliminar una ramificación del repositorio local denominado MyNewBranch:

```
git branch -D MyNewBranch
```

2. Para eliminar la rama del CodeCommit repositorio, ejecuta el `git push remote-name --delete branch-name` comando donde **remote-name** es el apodo que el repositorio local usa para el CodeCommit repositorio y **branch-name** es el nombre de la rama que deseas eliminar del CodeCommit repositorio.

 Tip

Para obtener una lista de los nombres de los CodeCommit repositorios y sus direcciones URL, ejecuta el `git remote -v` comando.

Por ejemplo, para eliminar una rama denominada MyNewBranch en el CodeCommit repositorio: `origin`

```
git push origin --delete MyNewBranch
```

 Tip

Este comando no elimina la ramificación si se trata de la ramificación predeterminada.

Para ver más opciones, consulte la documentación de Git.

Trabajo con las preferencias del usuario

Puede utilizar la consola AWS CodeCommit para configurar algunos ajustes predeterminados. Por ejemplo, puede cambiar sus preferencias para ver los cambios de código ya sea en línea o en una vista dividida. Al realizar un cambio en uno de estos ajustes, la consola AWS CodeCommit establece una cookie en el navegador que se almacena de forma automática y aplica sus opciones cada vez que utilice la consola. Cada vez que accede a la consola AWS CodeCommit utilizando ese navegador, estas preferencias se aplican a todos los repositorios de todas las regiones. Estas preferencias de configuración no son específicas del repositorio ni de cada región. No tienen ningún efecto en sus interacciones con la AWS CLI, la API de AWS CodeCommit ni otros servicios que interactúan con AWS CodeCommit.

Note

Las cookies de preferencias del usuario son específicas de un navegador. Si borra las cookies del navegador, sus preferencias se eliminan. Del mismo modo, si utiliza otro navegador para acceder a un repositorio, ese navegador no tiene acceso a las cookies de los otros navegadores. Sus preferencias no se mantienen.

Entre las preferencias del usuario se incluyen:

- Cuando se muestran los cambios en el código, puede elegir la vista Unified o Split y si desea mostrar u ocultar los cambios en los espacios en blanco.
- Al ver, editar o crear código, si se debe usar un fondo claro o un fondo oscuro en la ventana del editor de código.

No hay ninguna página para configurar sus preferencias. En su lugar, siempre que cambie una preferencia en la consola, como, por ejemplo, cómo ver los cambios de código, ese cambio se guarda y se aplica siempre que corresponda.

Migración a AWS CodeCommit

Puede migrar un repositorio de Git a un repositorio de CodeCommit de varias formas: puede clonarlo, crear un reflejo, migrar todas o solo algunas de sus ramificaciones, etc. También puede migrar el contenido local sin versión del equipo a CodeCommit.

Los siguientes temas muestran algunas de las formas que puede utilizar para migrar un repositorio. Los pasos podrían variar función del tipo, estilo o complejidad del repositorio, así como de las decisiones que tome sobre qué va a migrar y cómo va a hacerlo. En el caso de repositorios de gran tamaño, es posible que desee considerar la [migración incremental](#).

Note

Puede migrar a CodeCommit desde otros sistemas de control de versiones, como, por ejemplo, Perforce, Subversion o TFS, pero debe migrar a Git en primer lugar.

Para ver más opciones, consulte la documentación de Git.

También puede consultar la información sobre [la migración a Git](#) en el libro Pro Git de Scott Chacon y Ben Straub.

Temas

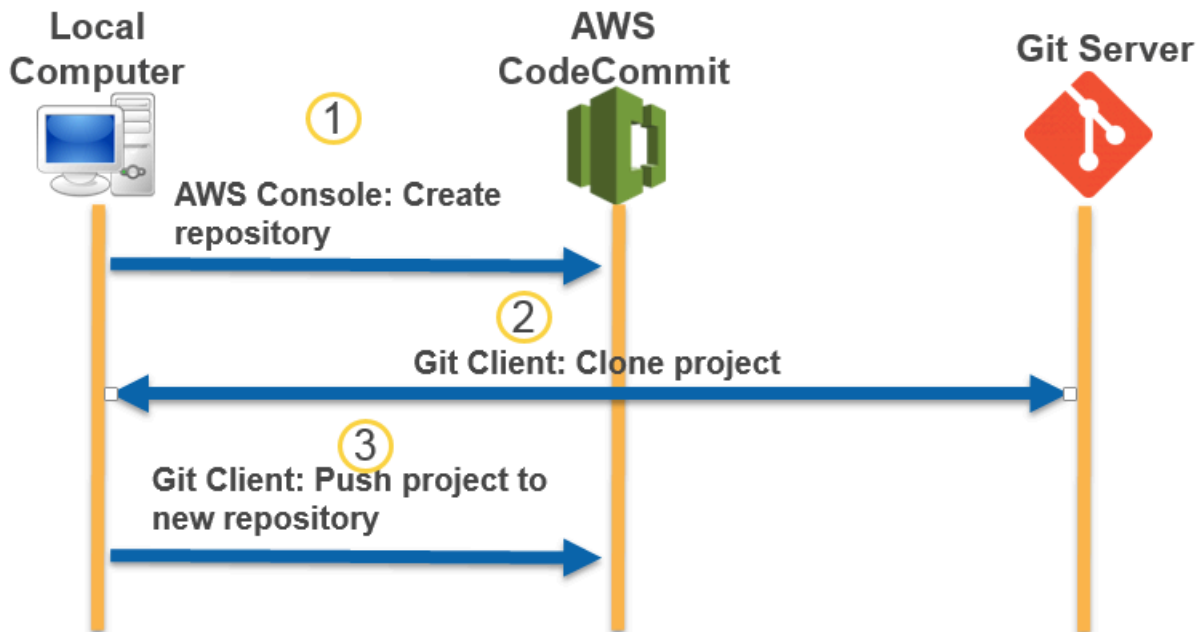
- [Migrar un repositorio de Git a AWS CodeCommit](#)
- [Migre contenido local o no versionado a AWS CodeCommit](#)
- [Migrar de forma incremental un repositorio](#)

Migrar un repositorio de Git a AWS CodeCommit

Puedes migrar un repositorio de Git existente a un CodeCommit repositorio. Los procedimientos incluidos en este tema muestran cómo migrar a CodeCommit un proyecto que se aloja en otro repositorio Git. Como parte de este proceso:

- Complete la configuración inicial requerida para CodeCommit.
- Cree un CodeCommit repositorio.
- Clona el repositorio y envíalo a él CodeCommit.
- Vea los archivos del CodeCommit repositorio.

- Comparte el CodeCommit repositorio con tu equipo.



Temas

- [Paso 0: Se requiere una configuración para acceder a CodeCommit](#)
- [Paso 1: Crea un repositorio CodeCommit](#)
- [Paso 2: clona el repositorio y envíalo al CodeCommit repositorio](#)
- [Paso 3: Ver los archivos en CodeCommit](#)
- [Paso 4: Compartir el repositorio CodeCommit](#)

Paso 0: Se requiere una configuración para acceder a CodeCommit

Antes de poder migrar un repositorio a CodeCommit, debe crear y configurar un usuario de IAM para el acceso a su ordenador local CodeCommit y configurarlo. También deberá instalar la AWS CLI para administrar CodeCommit. Aunque puedes realizar la mayoría de CodeCommit las tareas sin él, AWS CLI ofrece flexibilidad a la hora de trabajar con Git en la línea de comandos o en la terminal.

Si ya lo tienes configurado CodeCommit, puedes pasar a [Paso 1: Crea un repositorio CodeCommit](#).

Para crear y configurar un usuario de IAM para acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.
2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUseru otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Para instalar y configurar AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con él CodeCommit desde la línea de comandos. Le recomendamos que instale la versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales congit-remote-codecommit.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplous-east-2. Cuando se le pregunte el formato de salida predeterminado, indique json. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
```

Default region name [None]: *Type a supported region for CodeCommit here, and then press Enter*

Default output format [None]: *Type json here, and then press Enter*

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)
- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o a un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- sa-east-1
- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1

- us-gov-west-1
- us-gov-east-1
- eu-north-1
- ap-east-1
- me-south-1
- cn-north-1
- cn-northwest-1
- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

A continuación, deberá instalar Git.

- Para Linux, macOS o Unix:

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Note

Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a su forma de trabajar con CodeCommit ella. Si

Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

- Para Windows:

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git for Windows](#). Si utiliza este enlace para instalar Git, puede aceptar todos los ajustes predeterminados de la instalación excepto los siguientes:

- Cuando se le pida durante el paso Ajustar su entorno PATH, seleccione la opción de usar Git desde la línea de comandos.
- (Opcional) Si pretendes usar HTTPS con el asistente de credenciales que viene incluido en el en AWS CLI lugar de configurar las credenciales de Git para CodeCommit, en la página Configuración de opciones adicionales, asegúrate de que la opción Habilitar el administrador de credenciales de Git esté desactivada. El administrador de credenciales de Git solo es compatible CodeCommit si los usuarios de IAM configuran las credenciales de Git. Para obtener más información, consulte [Para usuarios HTTPS mediante credenciales de Git](#) y [Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio \(403\)](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. Ocasionalmente, un cambio en una función puede afectar a la forma en que funciona. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

CodeCommit admite la autenticación HTTPS y SSH. Para completar la configuración, debes configurar las credenciales de Git para CodeCommit (HTTPS, recomendado para la mayoría de los usuarios), un par de claves SSH para usar al acceder CodeCommit (SSH), git-remote-codecommit (recomendado para los usuarios que usan acceso federado) o el asistente de credenciales incluido en el (HTTPS). AWS CLI

- Para las credenciales de Git en todos los sistemas operativos compatibles, consulte [Paso 3: Crear credenciales de Git para las conexiones HTTPS a CodeCommit](#).
- Para SSH en Linux, macOS o Unix, consulte [SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit](#).
- Para SSH en Windows, consulte [Paso 3: Configura las claves públicas y privadas para Git y CodeCommit](#).
- Para git-remote-codecommit, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).
- Para ver el ayudante de credenciales en Linux, macOS o Unix, consulte [Configurar el ayudante de credenciales \(Linux, macOS o Unix\)](#).
- Para el auxiliar de credenciales en Windows, consulte [Configuración del auxiliar de credenciales \(Windows\)](#).

Paso 1: Crea un repositorio CodeCommit

En esta sección, utilizará la CodeCommit consola para crear el CodeCommit repositorio que utilizará para el resto de este tutorial. Para usar la AWS CLI para crear el repositorio, consulte [Crea un repositorio \(AWS CLI\)](#).

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elige la ubicación en la Región de AWS que deseas crear el repositorio. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositorios, seleccione Crear repositorio.
4. En la página Create repository (Crear repositorio), en el campo Repository name (Nombre de repositorio), escriba un nombre para el repositorio.

Note

Los nombres de repositorio distinguen entre mayúsculas y minúsculas. El nombre debe ser único en la Región de AWS y de su cuenta de Amazon Web Services.

5. (Opcional) En Descripción, introduzca una descripción para el repositorio. Esto puede ayudarle a usted y a otros usuarios a conocer el propósito del repositorio.

 Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si es un desarrollador de aplicaciones que utiliza GetRepository o BatchGetRepositories APIs y tiene previsto mostrar el campo de descripción del repositorio en un navegador web, consulte la [referencia de la CodeCommit API](#).

6. (Opcional) Selecciona Añadir etiqueta para añadir una o más etiquetas de repositorio (una etiqueta de atributo personalizada que te ayuda a organizar y gestionar tus AWS recursos) a tu repositorio. Para obtener más información, consulte [Etiquetar repositorios en AWS CodeCommit](#).
7. (Opcional) Amplíe la configuración adicional para especificar si desea utilizar la clave predeterminada Clave administrada de AWS o la propia clave gestionada por el cliente para cifrar y descifrar los datos de este repositorio. Si decide utilizar su propia clave gestionada por el cliente, debe asegurarse de que esté disponible en el Región de AWS lugar donde vaya a crear el repositorio y de que la clave esté activa. Para obtener más información, consulte [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#).
8. (Opcional) Seleccione Habilitar Amazon CodeGuru Reviewer para Java y Python si este repositorio contiene código Java o Python y desea que CodeGuru Reviewer lo analice. CodeGuru Reviewer utiliza varios modelos de aprendizaje automático para detectar defectos en el código y sugerir mejoras y correcciones en las solicitudes de incorporación de cambios. Para obtener más información, consulte la [Guía del usuario de Amazon CodeGuru Reviewer](#).
9. Seleccione Crear.

[Developer Tools](#) > [CodeCommit](#) > [Repositories](#) > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

100 characters maximum. Other limits apply.

createRepository.form.field.repositoryDescription.label - optional

1,000 characters maximum

[Cancel](#) [Create](#)

Una vez creado, el repositorio aparece en la lista Repositories (Repositorios). En la columna URL, seleccione el icono de copia y, a continuación, elija el protocolo (SSH o HTTPS) que se utiliza para conectar a CodeCommit. Copie la dirección URL.

Por ejemplo, si has asignado un nombre a tu repositorio *MyClonedRepository* y utilizas las credenciales de Git con HTTPS en la región EE.UU. Este (Ohio), la URL tendrá el siguiente aspecto:

```
https://git-codecommit.us-east-2.amazonaws.com/MyClonedRepository
```

Necesita esta URL más adelante en [Paso 2: clona el repositorio y envíalo al CodeCommit repositorio](#).

Paso 2: clona el repositorio y envíalo al CodeCommit repositorio

En esta sección, clona un repositorio Git en el equipo local, lo que dará lugar a lo que se denomina un repositorio local. A continuación, insertas el contenido del repositorio local en el CodeCommit repositorio que creaste anteriormente.

1. Desde la terminal o la línea de comandos de su equipo local, ejecute el git clone comando con la `--mirror` opción de clonar una copia simple del repositorio remoto en una nueva carpeta denominada `aws-codecommit-demo`. Se trata de un repositorio vacío concebido solamente para la migración. No es el repositorio local para interactuar con el repositorio migrado. CodeCommit Puede crearlo más adelante, una vez CodeCommit finalizada la migración.

En el siguiente ejemplo, se clona una aplicación de demostración alojada en GitHub (<https://github.com/awslabs/aws-demo-php-simple-app.git>) en un repositorio local de un directorio denominado `aws-codecommit-demo`.

```
git clone --mirror https://github.com/awslabs/aws-demo-php-simple-app.git aws-codecommit-demo
```

2. Cambie los directorios por el directorio en el que ha realizado la clonación.

```
cd aws-codecommit-demo
```

3. Ejecute el git push comando y especifique la URL y el nombre del CodeCommit repositorio de destino y la `--all` opción. (Esta es la URL que copió en [Paso 1: Crea un repositorio CodeCommit](#)).

Por ejemplo, si has asignado un nombre a tu repositorio `MyClonedRepository` y estás configurado para usar HTTPS, debes ejecutar el siguiente comando:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --all
```

Note

La opción `--all` solo envía todas las ramificaciones al repositorio. No envía otras referencias, como, por ejemplo, etiquetas. Si desea enviar etiquetas, espere hasta que se haya completado el envío inicial y, a continuación, envíe de nuevo, esta vez con la opción `--tags`.

```
git push ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --tags
```

Para obtener más información, consulte [Git push](#) en el sitio web de Git. Para obtener más información sobre el envío de grandes repositorios, especialmente cuando se

envían todas las referencias a la vez (por ejemplo, con la opción `--mirror`), consulte [Migrar de forma incremental un repositorio](#).

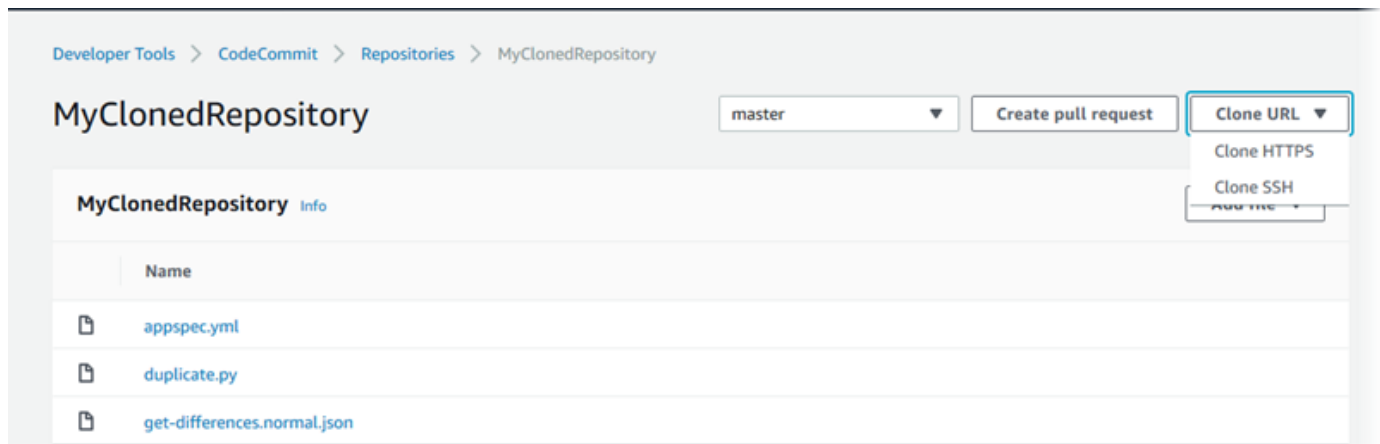
Puede eliminar la `aws-codecommit-demo` carpeta y su contenido después de migrar el repositorio a CodeCommit. Para crear un repositorio local con todas las referencias correctas para trabajar con el repositorio CodeCommit, ejecuta el `git clone` comando sin la `--mirror` opción:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository
```

Paso 3: Ver los archivos en CodeCommit

Una vez que haya introducido el contenido de su directorio, podrá utilizar la CodeCommit consola para ver rápidamente todos los archivos de ese repositorio.

1. Abre la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositorios, elige el nombre del repositorio (por ejemplo,). `MyClonedRepository`
3. Consulta los archivos del repositorio para ver las ramas, el clon URLs, la configuración y mucho más.



Paso 4: Compartir el repositorio CodeCommit

Al crear un repositorio en CodeCommit, se generan dos puntos de conexión: uno para las conexiones HTTPS y otro para las conexiones SSH. Ambas proporcionan conexiones seguras a través de una red. Los usuarios pueden utilizar cualquiera de los dos protocolos. Ambos puntos de enlace permanecen activos, independientemente del protocolo que recomiende a los usuarios. Antes

de poder compartir sus repositorios con otros usuarios, debe crear políticas de IAM que permitan a otros usuarios acceder a su repositorio. Proporcione las instrucciones de acceso a sus usuarios.

Creación de una política administrada por el cliente para su repositorio

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
2. En el área de navegación Dashboard, elija Políticas y, a continuación, seleccione Create Policy.
3. En la página Crear política, seleccione Importar política administrada.
4. En la página Importar políticas administradas, en Filtrar políticas, introduzca **AWSCodeCommitPowerUser**. Seleccione el botón situado junto al nombre de la política y, a continuación, seleccione Importar.
5. En la página Crear política, elija JSON. Sustituya la parte «*» de la Resource línea correspondiente a CodeCommit las acciones por el nombre de recurso de Amazon (ARN) del CodeCommit repositorio, como se muestra a continuación:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

 Tip

Para buscar el ARN del CodeCommit repositorio, vaya a la CodeCommit consola, elija el nombre del repositorio de la lista y, a continuación, elija Configuración. Para obtener más información, consulte [Visualización de la información de los repositorios](#).

Si desea que esta política se aplique a más de un repositorio, añada cada repositorio como recurso y especifique su ARN. Añada una coma entre cada declaración de recursos, tal y como se muestra aquí:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Cuando haya terminado de editar, seleccione Revisar la política.

6. En la página de revisión de la política, en Nombre, introduzca un nombre nuevo para la política (por ejemplo, *AWSCodeCommitPowerUser-MyDemoRepo*). Opcionalmente, puede proporcionar una descripción opcional para esta política.
7. Seleccione Crear política.

Para administrar el acceso a su repositorio, cree un grupo para los usuarios de IAM, añada usuarios de IAM a este grupo y adjunte la política administrada por el cliente que haya creado en el paso anterior. Adjunte cualquier otra política necesaria para el acceso, como IAMUser SSHKeys o IAMSelfManageServiceSpecificCredentials.

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el área de navegación Dashboard, elija Groups y, a continuación, seleccione Create New Group.
3. En la página Set Group Name (Establecer nombre del grupo), en Group Name (Nombre del grupo), introduzca un nombre para el grupo (por ejemplo, *MyDemoRepoGroup*) y seleccione Next Step (Siguiendo paso). Tenga en cuenta la posibilidad de incluir el nombre del repositorio como parte del nombre del grupo.

 Note

Este nombre debe ser exclusivo de una cuenta de Amazon Web Services.

4. Selecciona la casilla situada junto a la política gestionada por el cliente que creaste en la sección anterior (por ejemplo, *AWSCodeCommitPowerUser-MyDemoRepo*).
5. En la página Review, elija Create Group. IAM crea este grupo con las políticas especificadas ya adjuntas. El grupo aparece en la lista de grupos asociados con su cuenta de Amazon Web Services.
6. Seleccione su grupo de la lista.
7. En la página de resumen de grupo, elija la pestaña Users y, continuación, seleccione Add Users to Group. En la lista que muestra todos los usuarios asociados a tu cuenta de Amazon Web Services, selecciona las casillas situadas junto a los usuarios a los que quieres permitir el acceso al CodeCommit repositorio y, a continuación, selecciona Añadir usuarios.

 Tip

Puede usar el cuadro de búsqueda para encontrar rápidamente a los usuarios por su nombre.

8. Cuando haya agregado sus usuarios, cierre la consola de IAM.

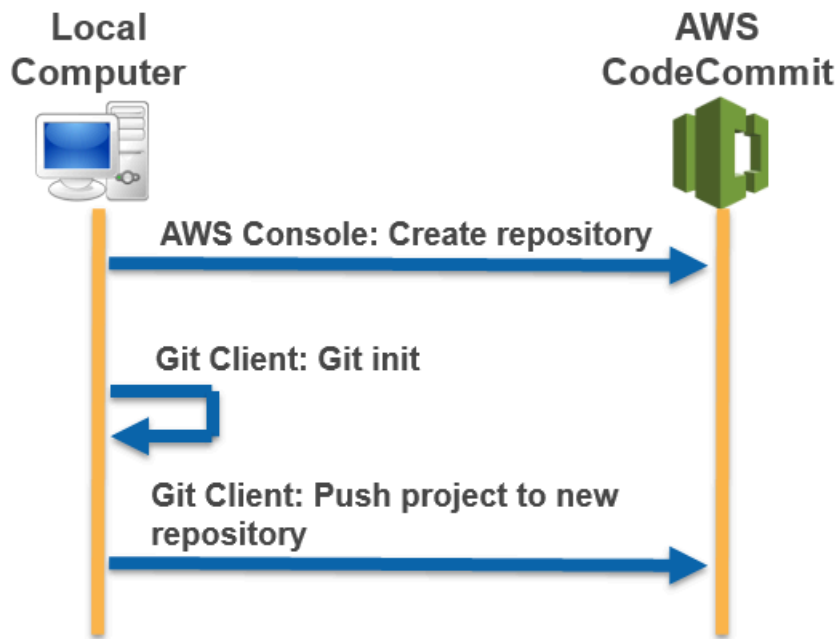
Una vez que haya creado un usuario de IAM al que pueda acceder CodeCommit mediante el grupo de políticas y las políticas que configuró, envíe a ese usuario la información necesaria para conectarse al repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositories (Repositorios), elija el repositorio que desea compartir.
4. En Clone URL (URL de clonación), elija el protocolo que desee que utilicen los usuarios. Esto copiará la URL de clonación del protocolo de conexión.
5. Envía a tus usuarios la URL del clon junto con cualquier otra instrucción, como instalar AWS CLI, configurar un perfil o instalar Git. Asegúrese de incluir la información de configuración para el protocolo de conexión (por ejemplo, HTTPS).

Migre contenido local o no versionado a AWS CodeCommit

Los procedimientos de este tema muestran cómo migrar un proyecto o contenido local existente en su equipo a un CodeCommit repositorio. Como parte de este proceso:

- Complete la configuración inicial necesaria para CodeCommit.
- Cree un CodeCommit repositorio.
- Coloca una carpeta local bajo el control de versiones de Git y envía el contenido de esa carpeta al CodeCommit repositorio.
- Visualiza los archivos del CodeCommit repositorio.
- Comparte el CodeCommit repositorio con tu equipo.



Temas

- [Paso 0: Se requiere una configuración para acceder a CodeCommit](#)
- [Paso 1: Crea un repositorio CodeCommit](#)
- [Paso 2: migra el contenido local al CodeCommit repositorio](#)
- [Paso 3: Ver los archivos en CodeCommit](#)
- [Paso 4: Comparte el CodeCommit repositorio](#)

Paso 0: Se requiere una configuración para acceder a CodeCommit

Antes de poder migrar el contenido local a CodeCommit, debe crear y configurar un usuario de IAM para el acceso al equipo local CodeCommit y configurarlo. También deberá instalar la AWS CLI para administrar CodeCommit. Aunque puedes realizar la mayoría de CodeCommit las tareas sin él, AWS CLI ofrece flexibilidad a la hora de trabajar con Git.

Si ya lo tienes configurado CodeCommit, puedes pasar a [Paso 1: Crea un repositorio CodeCommit](#).

Para crear y configurar un usuario de IAM para acceder CodeCommit

1. Para crear una cuenta de Amazon Web Services, acceda a <http://aws.amazon.com> y seleccione Registrarse.

2. Cree un usuario de IAM o use uno existente en su cuenta de Amazon Web Services. Asegúrese de que tiene un ID de clave de acceso y una clave de acceso secreta asociados a ese usuario de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de Amazon Web Services](#).

 Note

CodeCommit requiere AWS Key Management Service. Si utiliza un usuario de IAM existente, asegúrese de que no haya políticas adjuntas al usuario que denieguen expresamente las AWS KMS acciones requeridas por CodeCommit él. Para obtener más información, consulte [AWS KMS y cifrado](#).

3. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
4. En la consola de IAM, en el panel de navegación, elija Usuarios y, a continuación, elija el usuario de IAM que desee configurar para el acceso. CodeCommit
5. En la pestaña Permissions, seleccione Add Permissions.
6. En Grant permissions, elija Attach existing policies directly.
7. En la lista de políticas, seleccione AWSCodeCommitPowerUser otra política gestionada de CodeCommit acceso. Para obtener más información, consulte [AWS políticas gestionadas para CodeCommit](#).

Una vez que haya seleccionado la política que desee asociar, seleccione Siguiente: Revisar para revisar la lista de políticas que se van a asociar al usuario de IAM. Si la lista es correcta, seleccione Add permissions.

Para obtener más información sobre las políticas CodeCommit administradas y sobre cómo compartir el acceso a los repositorios con otros grupos y usuarios, consulte [Compartir un repositorio](#) y [Autenticación y control de acceso para AWS CodeCommit](#).

Para instalar y configurar el AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con él CodeCommit desde la línea de comandos. Le recomendamos que instale la versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales congit-remote-codecommit.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

 Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplous-east-2. Cuando se le pregunte el formato de salida predeterminado, indique json. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
Default region name [None]: Type a supported region for CodeCommit here, and then press Enter
Default output format [None]: Type json here, and then press Enter
```

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)
- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- sa-east-1
- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1
- ca-central-1
- us-gov-west-1
- us-gov-east-1
- eu-north-1
- ap-east-1

- cn-north-1
- cn-northwest-1
- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

A continuación, deberá instalar Git.

- Para Linux, macOS o Unix:

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git Downloads](#).

Note

Git es una plataforma en evolución que se actualiza periódicamente. En ocasiones, un cambio en una función puede afectar a su funcionamiento. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

- Para Windows:

Para trabajar con archivos, confirmaciones y otra información de los CodeCommit repositorios, debes instalar Git en tu máquina local. CodeCommit es compatible con las versiones 1.7.9 y

posteriores de Git. La versión 2.28 de Git admite la configuración del nombre de la ramificación para las confirmaciones iniciales. Recomendamos usar una versión reciente de Git.

Para instalar Git, le recomendamos sitios web como [Git for Windows](#). Si utiliza este enlace para instalar Git, puede aceptar todos los ajustes predeterminados de la instalación excepto los siguientes:

- Cuando se le pida durante el paso Ajustar su entorno PATH, seleccione la opción de usar Git desde la línea de comandos.
- (Opcional) Si pretendes usar HTTPS con el asistente de credenciales que viene incluido en el en AWS CLI lugar de configurar las credenciales de Git para CodeCommit, en la página Configuración de opciones adicionales, asegúrate de que la opción Habilitar el administrador de credenciales de Git esté desactivada. El administrador de credenciales de Git solo es compatible CodeCommit si los usuarios de IAM configuran las credenciales de Git. Para obtener más información, consulte [Para usuarios HTTPS mediante credenciales de Git](#) y [Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio \(403\)](#).

 Note

Git es una plataforma en evolución que se actualiza periódicamente. En ocasiones, un cambio en una función puede afectar a su funcionamiento. CodeCommit Si tienes problemas con una versión específica de Git CodeCommit, consulta la información en [Solución de problemas](#).

CodeCommit admite la autenticación HTTPS y SSH. Para completar la configuración, debes configurar las credenciales de Git para CodeCommit (HTTPS, recomendado para la mayoría de los usuarios), un par de claves SSH (SSH) para usar al acceder CodeCommit git-remote-codecommit (recomendado para los usuarios que usan acceso federado) o el asistente de credenciales incluido en el. AWS CLI

- Para las credenciales de Git en todos los sistemas operativos compatibles, consulte [Paso 3: Crear credenciales de Git para las conexiones HTTPS a CodeCommit](#).
- Para SSH en Linux, macOS o Unix, consulte [SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit](#).
- Para SSH en Windows, consulte [Paso 3: Configura las claves públicas y privadas para Git y CodeCommit](#).

- Para git-remote-codecommit, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).
- Para ver el ayudante de credenciales en Linux, macOS o Unix, consulte [Configurar el ayudante de credenciales \(Linux, macOS o Unix\)](#).
- Para el auxiliar de credenciales en Windows, consulte [Configuración del auxiliar de credenciales \(Windows\)](#).

Paso 1: Crea un repositorio CodeCommit

En esta sección, utilizará la CodeCommit consola para crear el CodeCommit repositorio que utilizará para el resto de este tutorial. Para usar la AWS CLI para crear el repositorio, consulte [Crea un repositorio \(AWS CLI\)](#).

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elige Región de AWS dónde quieres crear el repositorio. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositorios, seleccione Crear repositorio.
4. En la página Create repository (Crear repositorio), en el campo Repository name (Nombre de repositorio), escriba un nombre para el repositorio.

Note

Los nombres de repositorio distinguen entre mayúsculas y minúsculas. El nombre debe ser único en la Región de AWS y de su cuenta de Amazon Web Services.

5. (Opcional) En Descripción, introduzca una descripción para el repositorio. Esto puede ayudarle a usted y a otros usuarios a conocer el propósito del repositorio.

Note

El campo de descripción muestra Marcado en la consola y acepta todos los caracteres HTML y caracteres Unicode válidos. Si es un desarrollador de aplicaciones que utiliza GetRepository o BatchGetRepositories APIs y tiene previsto mostrar el campo de descripción del repositorio en un navegador web, consulte la [referencia de la CodeCommit API](#).

6. (Opcional) Seleccione Añadir etiqueta para añadir una o más etiquetas de repositorio (una etiqueta de atributo personalizada que te ayuda a organizar y gestionar tus AWS recursos) a tu repositorio. Para obtener más información, consulte [Etiquetar repositorios en AWS CodeCommit](#).
7. (Opcional) Amplíe la configuración adicional para especificar si desea utilizar la clave predeterminada Clave administrada de AWS o la propia clave gestionada por el cliente para cifrar y descifrar los datos de este repositorio. Si decide utilizar su propia clave gestionada por el cliente, debe asegurarse de que esté disponible en el Región de AWS lugar donde vaya a crear el repositorio y de que la clave esté activa. Para obtener más información, consulte [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#).
8. (Opcional) Seleccione Habilitar Amazon CodeGuru Reviewer para Java y Python si este repositorio contiene código Java o Python y desea que CodeGuru Reviewer lo analice. CodeGuru Reviewer utiliza varios modelos de aprendizaje automático para detectar defectos en el código y sugerir mejoras y correcciones en las solicitudes de incorporación de cambios. Para obtener más información, consulte la [Guía del usuario de Amazon CodeGuru Reviewer](#).
9. Seleccione Crear.

Una vez creado, el repositorio aparece en la lista Repositories (Repositorios). En la columna URL, seleccione el icono de copia y, a continuación, elija el protocolo (HTTPS o SSH) que se utiliza para conectar a CodeCommit. Copie la dirección URL.

Por ejemplo, si has asignado un nombre a tu repositorio *MyFirstRepo* y utilizas HTTPS, la URL tendría el siguiente aspecto:

```
https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyFirstRepo
```

Necesita esta URL más adelante en [Paso 2: migra el contenido local al CodeCommit repositorio](#).

Paso 2: migra el contenido local al CodeCommit repositorio

Ahora que tienes un CodeCommit repositorio, puedes elegir un directorio en tu computadora local para convertirlo en un repositorio Git local. El comando `git init` se puede utilizar para convertir el contenido sin versión existente en un repositorio de Git o, si aún no dispone de archivos o contenido, para inicializar un nuevo repositorio vacío.

1. En el terminal o en la línea de comandos del equipo local, cambie los directorios al directorio que desee utilizar como origen del repositorio.

2. Ejecute el siguiente comando para configurar Git para que use una ramificación predeterminada llamada **main**:

```
git config --local init.defaultBranch main
```

También puede ejecutar este comando para establecer como nombre de ramificación predeterminado todos los repositorios de **main** recién creados:

```
git config --global init.defaultBranch main
```

3. Ejecute el comando `git init` para inicializar el control de versiones de Git en el directorio. En la raíz del directorio se crea un subdirectorio `.git` que permite realizar un seguimiento del control de versiones. La carpeta `.git` también contiene todos los metadatos que necesita el repositorio.

```
git init
```

4. Compruebe el estado del directorio inicializado ejecutando el siguiente comando:

```
git status
```

Añada los archivos que desea incluir en el control de versiones. En este tutorial, ejecuta el comando `git add` con el especificador `.` para añadir todos los archivos de este directorio. Para ver otras opciones, consulte la documentación de Git.

```
git add .
```

5. Cree una confirmación para añadir archivos con un mensaje de confirmación.

```
git commit -m "Initial commit"
```

6. Ejecuta el `git push` comando especificando la URL y el nombre del CodeCommit repositorio de destino y la `--all` opción. (Esta es la URL que copió en [Paso 1: Crea un repositorio CodeCommit](#)).

Por ejemplo, si has asignado un nombre a tu repositorio *MyFirstRepo* y estás configurado para usar HTTPS, debes ejecutar el siguiente comando:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyFirstRepo --all
```

Paso 3: Ver los archivos en CodeCommit

Después de introducir el contenido de su directorio, puede utilizar la CodeCommit consola para ver rápidamente todos los archivos del repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En Repositorios, elija el nombre del repositorio (por ejemplo, *MyFirstRepository*) de la lista.
3. Vea los archivos del repositorio para ver las ramas, el clon URLs, la configuración y más.

Paso 4: Comparte el CodeCommit repositorio

Al crear un repositorio en CodeCommit, se generan dos puntos de conexión: uno para las conexiones HTTPS y otro para las conexiones SSH. Ambas proporcionan conexiones seguras a través de una red. Los usuarios pueden utilizar cualquiera de los dos protocolos. Ambos puntos de enlace permanecen activos, independientemente del protocolo que recomiende a los usuarios. Antes de poder compartir sus repositorios con otros usuarios, debe crear políticas de IAM que permitan a otros usuarios acceder a su repositorio. Proporcione las instrucciones de acceso a sus usuarios.

Creación de una política administrada por el cliente para su repositorio

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. <https://console.aws.amazon.com/iam/>
2. En el área de navegación Dashboard, elija Políticas y, a continuación, seleccione Create Policy.
3. En la página Crear política, seleccione Importar política administrada.
4. En la página Importar políticas administradas, en Filtrar políticas, introduzca **AWSCodeCommitPowerUser**. Seleccione el botón situado junto al nombre de la política y, a continuación, seleccione Importar.
5. En la página Crear política, elija JSON. Sustituya la parte «*» de la Resource línea correspondiente a CodeCommit las acciones por el nombre de recurso de Amazon (ARN) del CodeCommit repositorio, como se muestra aquí:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

 Tip

Para buscar el ARN del CodeCommit repositorio, vaya a la CodeCommit consola, elija el nombre del repositorio de la lista y, a continuación, elija Configuración. Para obtener más información, consulte [Visualización de la información de los repositorios](#).

Si desea que esta política se aplique a más de un repositorio, añada cada repositorio como recurso y especifique su ARN. Añada una coma entre cada declaración de recursos, tal y como se muestra aquí:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Cuando haya terminado de editar, seleccione Revisar la política.

6. En la página de revisión de la política, en Nombre, introduzca un nombre nuevo para la política (por ejemplo, *AWSCodeCommitPowerUser-MyDemoRepo*). Opcionalmente, puede proporcionar una descripción opcional para esta política.
7. Seleccione Crear política.

Para administrar el acceso a su repositorio, cree un grupo para los usuarios de IAM, añada usuarios de IAM a este grupo y adjunte la política administrada por el cliente que haya creado en el paso anterior. Adjunte cualquier otra política necesaria para el acceso, como, por ejemplo, *IAMSelfManageServiceSpecificCredentials* o *IAMUserSSHKeys*.

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el área de navegación Dashboard, elija Groups y, a continuación, seleccione Create New Group.
3. En la página Set Group Name (Establecer nombre del grupo), en Group Name (Nombre del grupo), introduzca un nombre para el grupo (por ejemplo, *MyDemoRepoGroup*) y seleccione Next Step (Siguiendo paso). Tenga en cuenta la posibilidad de incluir el nombre del repositorio como parte del nombre del grupo.

 Note

Este nombre debe ser exclusivo de una cuenta de Amazon Web Services.

4. Selecciona la casilla situada junto a la política gestionada por el cliente que creaste en la sección anterior (por ejemplo, AWSCodeCommitPowerUser-MyDemoRepo).
5. En la página Review, elija Create Group. IAM crea este grupo con las políticas especificadas ya adjuntas. El grupo aparece en la lista de grupos asociados con su cuenta de Amazon Web Services.
6. Seleccione su grupo de la lista.
7. En la página de resumen de grupo, elija la pestaña Users y, continuación, seleccione Add Users to Group. En la lista que muestra todos los usuarios asociados a tu cuenta de Amazon Web Services, selecciona las casillas situadas junto a los usuarios a los que quieres permitir el acceso al CodeCommit repositorio y, a continuación, selecciona Añadir usuarios.

 Tip

Puede usar el cuadro de búsqueda para encontrar rápidamente a los usuarios por su nombre.

8. Cuando haya agregado sus usuarios, cierre la consola de IAM.

Una vez que haya creado un usuario de IAM para utilizarlo como acceso CodeCommit mediante el grupo de políticas y las políticas que configuró, envíe a ese usuario la información necesaria para conectarse al repositorio.

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. En el selector de regiones, elija el Región de AWS lugar donde se creó el repositorio. Los repositorios son específicos de un Región de AWS. Para obtener más información, consulte [Regiones y puntos de conexión de Git](#).
3. En la página Repositories (Repositorios), elija el repositorio que desea compartir.
4. En Clone URL (URL de clonación), elija el protocolo que desee que utilicen los usuarios. Esto copiará la URL de clonación del protocolo de conexión.

5. Envía a tus usuarios la URL del clon junto con cualquier otra instrucción, como instalar AWS CLI, configurar un perfil o instalar Git. Asegúrese de incluir la información de configuración para el protocolo de conexión (por ejemplo, HTTPS).

Migrar de forma incremental un repositorio

Al migrar a AWS CodeCommit, considere la posibilidad de enviar su repositorio en incrementos o fragmentos para reducir las posibilidades de que un problema de red intermitente o una degradación del rendimiento de la red provoque un error en toda la transferencia. Al realizar envíos incrementales con un script como el incluido aquí, puede reiniciar la migración y enviar únicamente aquellas confirmaciones que no se hayan podido enviar en el primer intento.

Los procedimientos incluidos en este tema explican cómo crear y ejecutar un script que migre su repositorio de forma incremental y envíe de nuevo solo aquellos fragmentos que no se han podido enviar hasta que se complete toda la migración.

Estas instrucciones se han redactado presuponiendo que ya ha completado los pasos indicados en [Configuración](#) y [Creación de un repositorio](#).

Temas

- [Paso 0: Determinar si realizar una migración de forma incremental](#)
- [Paso 1: Instale los requisitos previos y añada el CodeCommit repositorio como remoto](#)
- [Paso 2: Crear el script para migrar de forma incremental](#)
- [Paso 3: Ejecute el script y migre de forma incremental a CodeCommit](#)
- [Anexo: script de muestra incremental-repo-migration.py](#)

Paso 0: Determinar si realizar una migración de forma incremental

A la hora de determinar el tamaño total de su repositorio y decidir si migrar de forma incremental, debe considerar determinados factores. El factor más evidente es el tamaño total de los elementos del repositorio. Los factores, como el historial acumulado del repositorio, también pueden contribuir al tamaño. Un repositorio con años de historia y ramificaciones puede ser muy grande, aunque sus elementos individuales no lo sean. Puede utilizar distintas estrategias para simplificar la migración de estos repositorios y que sea más eficaz. Por ejemplo, puede utilizar una estrategia de clonación superficial para clonar un repositorio con un largo historial de desarrollo o puede desactivar la compresión delta para archivos binarios grandes. Puede buscar otras opciones en la documentación

de Git o puede optar por establecer y configurar envíos incrementales para migrar el repositorio utilizando el script de muestra incluido en este tema `incremental-repo-migration.py`.

Es posible que le interese configurar envíos incrementales en caso de que se cumpla una o más de las siguientes condiciones:

- El repositorio que desea migrar tiene más de cinco años de historia.
- Su conexión a Internet está sometida a interrupciones intermitentes, pérdidas de paquetes, respuestas lentas o cualquier otra interrupción del servicio.
- El tamaño total del repositorio es superior a 2 GB e intenta migrar todo el repositorio.
- El repositorio contiene elementos grandes o binarios que no se comprimen bien, como archivos de imágenes grandes con más de cinco versiones controladas.
- Ya has intentado realizar una migración CodeCommit y has recibido un mensaje de «error de servicio interno».

Aunque no se cumpla ninguna de las condiciones anteriores, también puede optar por enviar de forma incremental.

Paso 1: Instale los requisitos previos y añada el CodeCommit repositorio como remoto

Puede crear su propio script personalizado con sus propios requisitos previos. Si utiliza el ejemplo incluido en este tema, debe:

- Instalar los requisitos previos.
- Clonar el repositorio en su equipo local.
- Añadir el CodeCommit repositorio como remoto al repositorio que desee migrar.

Configurado para ejecutar `incremental-repo-migration.py`

1. En el equipo local, instale Python 2.6 o posterior. Para obtener más información y las versiones más recientes, consulte [el sitio web de Python](#).
2. En la misma computadora, instale GitPython, que es una biblioteca de Python que se usa para interactuar con los repositorios de Git. Para obtener más información, consulte la [documentación de GitPython](#).

3. Utilice el comando `git clone --mirror` para clonar el repositorio que desea migrar a su equipo local. Desde el terminal (Linux, macOS o Unix) o el símbolo del sistema (Windows), utilice el comando `git clone --mirror` para crear un repositorio local para el repositorio, incluido el directorio en el que desea crear el repositorio local. Por ejemplo, para clonar un repositorio de Git llamado *MyMigrationRepo* con una URL *https://example.com/my-repo/* de en un directorio llamado *my-repo*:

```
git clone --mirror https://example.com/my-repo/MyMigrationRepo.git my-repo
```

El resultado debería ser similar al siguiente, que indica que el repositorio se ha clonado en un repositorio local vacío denominado *my-repo*:

```
Cloning into bare repository 'my-repo'...
remote: Counting objects: 20, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 20 (delta 5), reused 15 (delta 3)
Unpacking objects: 100% (20/20), done.
Checking connectivity... done.
```

4. Cambia los directorios al repositorio local del repositorio que acabas de clonar (por ejemplo, *my-repo*). En ese directorio, utilice el comando `git remote add` **DefaultRemoteName RemoteRepositoryURL** para añadir el repositorio de CodeCommit como repositorio remoto del repositorio local.

Note

Si envía repositorios de gran tamaño, considere la posibilidad de utilizar SSH en lugar de HTTPS. Si envía un cambio grande, un gran número de cambios o un repositorio grande, a menudo las conexiones HTTPS de ejecución prolongada suelen interrumpirse de forma prematura debido a problemas de red o de configuración del cortafuegos. Para obtener más información sobre la configuración de CodeCommit SSH, consulte o. [Para conexiones SSH en Linux, macOS o Unix](#) [Para conexiones SSH en Windows](#)

Por ejemplo, utilice el siguiente comando para añadir el punto de conexión SSH a un CodeCommit repositorio *MyDestinationRepo* denominado repositorio remoto para el nombre remoto: `codecommit`

```
git remote add codecommit ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDestinationRepo
```

i Tip

Dado que se trata de un clon, el nombre remoto predeterminado (`origin`) ya está en uso. Deberá utilizar otro nombre remoto. Aunque el ejemplo utiliza `codecommit`, puede usar el nombre que desee. Utilice el comando `git remote show` para revisar la lista de remotos configurados para su repositorio local.

5. Utilice el comando `git remote -v` para mostrar la búsqueda y el envío de la configuración de su repositorio local y confirmar que están configurados correctamente. Por ejemplo:

```
codecommit  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDestinationRepo
(fetch)
codecommit  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDestinationRepo
(push)
```

i Tip

Si siguen apareciendo entradas de búsqueda y envío para otro repositorio remoto (por ejemplo, entradas de origen), utilice el comando `git remote set-url --delete` para eliminarlas.

Paso 2: Crear el script para migrar de forma incremental

Estos pasos se han escrito partiendo del supuesto de que está utilizando el script de ejemplo `incremental-repo-migration.py`.

1. Abra un editor de texto y pegue el contenido [del script de muestra](#) en un documento vacío.
2. Guarde el documento en un directorio de documentos (no el directorio de trabajo de su repositorio local) y nómbrelo `incremental-repo-migration.py`. Asegúrese de que el directorio que elija esté configurado en sus variables del entorno local o de la ruta, para poder ejecutar el script de Python desde una línea de comandos o terminal.

Paso 3: Ejecute el script y migre de forma incremental a CodeCommit

Ahora que ha creado su `incremental-repo-migration.py` script, puede usarlo para migrar de forma incremental un repositorio local a un repositorio. CodeCommit De forma predeterminada, el script envía las confirmaciones en lotes de 1 000 e intenta utilizar la configuración de Git para el directorio desde el que se ejecuta como la configuración del repositorio local y del repositorio remoto. Puede utilizar las opciones incluidas en `incremental-repo-migration.py` para configurar otros ajustes, si es necesario.

1. Desde el terminal o la línea de comandos, cambie los directorios al repositorio local al que desea migrar.
2. Desde el directorio ejecute el siguiente comando:

```
python incremental-repo-migration.py
```

3. El script se ejecuta y muestra el progreso en el terminal o en la línea de comandos. El progreso de algunos repositorios grandes es más lento. El script se detiene si intenta enviarlo sin éxito tres veces seguidas. A continuación, podrá volver a ejecutar el script que empieza de nuevo desde el lote que ha devuelto un error. Podrá volver a ejecutar el script hasta que todo se envíe correctamente y se haya completado la migración.

Tip

Puede ejecutar `incremental-repo-migration.py` desde cualquier directorio siempre y cuando utilice las opciones `-l` y `-r` para especificar la configuración local y remota. Por ejemplo, si quieres usar el script de cualquier directorio para migrar un repositorio local ubicado en `/tmp/my-repo` a un remoto con el siguiente apodo: `codecommit`

```
python incremental-repo-migration.py -l "/tmp/my-repo" -r "codecommit"
```

También puede utilizar la opción `-b` para cambiar el tamaño del lote predeterminado utilizado para los envíos incrementales. Por ejemplo, si envía regularmente un repositorio con archivos binarios muy grandes que cambian con frecuencia y trabaja desde una ubicación con un ancho de banda de la red limitado, le recomendamos que utilice la opción `-b` para cambiar el tamaño de 1 000 a 500. Por ejemplo:

```
python incremental-repo-migration.py -b 500
```

De este modo, el repositorio local envía de forma incremental en lotes de 500 confirmaciones. Si decide cambiar el tamaño del lote de nuevo al migrar el repositorio (por ejemplo, si decide reducir el tamaño del lote después de un intento erróneo), recuerde utilizar la opción `-c` para eliminar las etiquetas del lote antes de restablecer el tamaño del lote con `-b`:

```
python incremental-repo-migration.py -c
python incremental-repo-migration.py -b 250
```

Important

No utilice la opción `-c` si desea volver a ejecutar el script después de un intento erróneo. La opción `-c` elimina las etiquetas que se utilizan para crear los lotes de confirmaciones. Utilice la opción `-c` solo si desea cambiar el tamaño del lote y empezar de nuevo, o si decide que ya no quiere utilizar el script.

Anexo: script de muestra **incremental-repo-migration.py**

Para su comodidad, hemos desarrollado un script de Python de muestra, `incremental-repo-migration.py`, para enviar un repositorio de forma incremental. Este script es un ejemplo de código abierto y se proporciona tal cual.

```
# Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved. Licensed
# under the Amazon Software License (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#   http://aws.amazon.com/asl/
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, express or implied. See the License for
# the specific language governing permissions and limitations under the License.

#!/usr/bin/env python

import os
import sys
from optparse import OptionParser
from git import Repo, TagReference, RemoteProgress, GitCommandError
```

```
class PushProgressPrinter(RemoteProgress):
    def update(self, op_code, cur_count, max_count=None, message=""):
        op_id = op_code & self.OP_MASK
        stage_id = op_code & self.STAGE_MASK
        if op_id == self.WRITING and stage_id == self.BEGIN:
            print("\t0bjects: %d" % max_count)

class RepositoryMigration:
    MAX_COMMITS_TOLERANCE_PERCENT = 0.05
    PUSH_RETRY_LIMIT = 3
    MIGRATION_TAG_PREFIX = "codecommit_migration_"

    def migrate_repository_in_parts(
        self, repo_dir, remote_name, commit_batch_size, clean
    ):
        self.next_tag_number = 0
        self.migration_tags = []
        self.walked_commits = set()
        self.local_repo = Repo(repo_dir)
        self.remote_name = remote_name
        self.max_commits_per_push = commit_batch_size
        self.max_commits_tolerance = (
            self.max_commits_per_push * self.MAX_COMMITS_TOLERANCE_PERCENT
        )

        try:
            self.remote_repo = self.local_repo.remote(remote_name)
            self.get_remote_migration_tags()
        except (ValueError, GitCommandError):
            print(
                "Could not contact the remote repository. The most common reasons for this error are that the name of the remote repository is incorrect, or that you do not have permissions to interact with that remote repository."
            )
            sys.exit(1)

        if clean:
            self.clean_up(clean_up_remote=True)
            return

        self.clean_up()
```

```
print("Analyzing repository")
head_commit = self.local_repo.head.commit
sys.setrecursionlimit(max(sys.getrecursionlimit(), head_commit.count()))

# tag commits on default branch
leftover_commits = self.migrate_commit(head_commit)
self.tag_commits([commit for (commit, commit_count) in leftover_commits])

# tag commits on each branch
for branch in self.local_repo.heads:
    leftover_commits = self.migrate_commit(branch.commit)
    self.tag_commits([commit for (commit, commit_count) in leftover_commits])

# push the tags
self.push_migration_tags()

# push all branch references
for branch in self.local_repo.heads:
    print("Pushing branch %s" % branch.name)
    self.do_push_with_retries(ref=branch.name)

# push all tags
print("Pushing tags")
self.do_push_with_retries(push_tags=True)

self.get_remote_migration_tags()
self.clean_up(clean_up_remote=True)

print("Migration to CodeCommit was successful")

def migrate_commit(self, commit):
    if commit in self.walked_commits:
        return []

    pending_ancestor_pushes = []
    commit_count = 1

    if len(commit.parents) > 1:
        # This is a merge commit
        # Ensure that all parents are pushed first
        for parent_commit in commit.parents:
            pending_ancestor_pushes.extend(self.migrate_commit(parent_commit))
    elif len(commit.parents) == 1:
```

```

        # Split linear history into individual pushes
        next_ancestor, commits_to_next_ancestor = self.find_next_ancestor_for_push(
            commit.parents[0]
        )
        commit_count += commits_to_next_ancestor
        pending_ancestor_pushes.extend(self.migrate_commit(next_ancestor))

    self.walked_commits.add(commit)

    return self.stage_push(commit, commit_count, pending_ancestor_pushes)

def find_next_ancestor_for_push(self, commit):
    commit_count = 0

    # Traverse linear history until we reach our commit limit, a merge commit, or
    # an initial commit
    while (
        len(commit.parents) == 1
        and commit_count < self.max_commits_per_push
        and commit not in self.walked_commits
    ):
        commit_count += 1
        self.walked_commits.add(commit)
        commit = commit.parents[0]

    return commit, commit_count

def stage_push(self, commit, commit_count, pending_ancestor_pushes):
    # Determine whether we can roll up pending ancestor pushes into this push
    combined_commit_count = commit_count + sum(
        ancestor_commit_count
        for (ancestor, ancestor_commit_count) in pending_ancestor_pushes
    )

    if combined_commit_count < self.max_commits_per_push:
        # don't push anything, roll up all pending ancestor pushes into this
        # pending push
        return [(commit, combined_commit_count)]

    if combined_commit_count <= (
        self.max_commits_per_push + self.max_commits_tolerance
    ):
        # roll up everything into this commit and push
        self.tag_commits([commit])

```

```

        return []

    if commit_count >= self.max_commits_per_push:
        # need to push each pending ancestor and this commit
        self.tag_commits(
            [
                ancestor
                for (ancestor, ancestor_commit_count) in pending_ancestor_pushes
            ]
        )
        self.tag_commits([commit])
        return []

    # push each pending ancestor, but roll up this commit
    self.tag_commits(
        [ancestor for (ancestor, ancestor_commit_count) in pending_ancestor_pushes]
    )
    return [(commit, commit_count)]

def tag_commits(self, commits):
    for commit in commits:
        self.next_tag_number += 1
        tag_name = self.MIGRATION_TAG_PREFIX + str(self.next_tag_number)

        if tag_name not in self.remote_migration_tags:
            tag = self.local_repo.create_tag(tag_name, ref=commit)
            self.migration_tags.append(tag)
        elif self.remote_migration_tags[tag_name] != str(commit):
            print(
                "Migration tags on the remote do not match the local tags. Most
likely your batch size has changed since the last time you ran this script. Please run
this script with the --clean option, and try again."
            )
            sys.exit(1)

def push_migration_tags(self):
    print("Will attempt to push %d tags" % len(self.migration_tags))
    self.migration_tags.sort(
        key=lambda tag: int(tag.name.replace(self.MIGRATION_TAG_PREFIX, ""))
    )
    for tag in self.migration_tags:
        print(
            "Pushing tag %s (out of %d tags), commit %s"
            % (tag.name, self.next_tag_number, str(tag.commit))
        )

```

```
)
self.do_push_with_retries(ref=tag.name)

def do_push_with_retries(self, ref=None, push_tags=False):
    for i in range(0, self.PUSH_RETRY_LIMIT):
        if i == 0:
            progress_printer = PushProgressPrinter()
        else:
            progress_printer = None

        try:
            if push_tags:
                infos = self.remote_repo.push(tags=True, progress=progress_printer)
            elif ref is not None:
                infos = self.remote_repo.push(
                    refspec=ref, progress=progress_printer
                )
            else:
                infos = self.remote_repo.push(progress=progress_printer)

            success = True
            if len(infos) == 0:
                success = False
            else:
                for info in infos:
                    if (
                        info.flags & info.UP_TO_DATE
                        or info.flags & info.NEW_TAG
                        or info.flags & info.NEW_HEAD
                    ):
                        continue
                    success = False
                    print(info.summary)

            if success:
                return
        except GitCommandError as err:
            print(err)

    if push_tags:
        print("Pushing all tags failed after %d attempts" %
              (self.PUSH_RETRY_LIMIT))
    elif ref is not None:
        print("Pushing %s failed after %d attempts" % (ref, self.PUSH_RETRY_LIMIT))
```

```

        print(
            "For more information about the cause of this error, run the following
command from the local repo: 'git push %s %s'"
            % (self.remote_name, ref)
        )
    else:
        print(
            "Pushing all branches failed after %d attempts"
            % (self.PUSH_RETRY_LIMIT)
        )
    sys.exit(1)

def get_remote_migration_tags(self):
    remote_tags_output = self.local_repo.git.ls_remote(
        self.remote_name, tags=True
    ).split("\n")
    self.remote_migration_tags = dict(
        (tag.split()[1].replace("refs/tags/", ""), tag.split()[0])
        for tag in remote_tags_output
        if self.MIGRATION_TAG_PREFIX in tag
    )

def clean_up(self, clean_up_remote=False):
    tags = [
        tag
        for tag in self.local_repo.tags
        if tag.name.startswith(self.MIGRATION_TAG_PREFIX)
    ]

    # delete the local tags
    TagReference.delete(self.local_repo, *tags)

    # delete the remote tags
    if clean_up_remote:
        tags_to_delete = [":" + tag_name for tag_name in
self.remote_migration_tags]
        self.remote_repo.push(refspec=tags_to_delete)

parser = OptionParser()
parser.add_option(
    "-l",
    "--local",
    action="store",

```



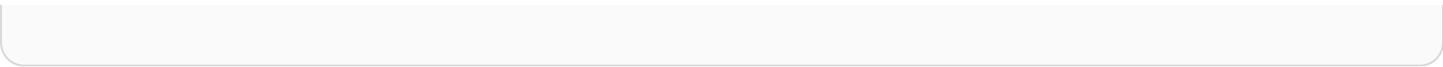
```

    dest="localrepo",
    default=os.getcwd(),
    help="The path to the local repo. If this option is not specified, the script will
attempt to use current directory by default. If it is not a local git repo, the script
will fail.",
)
parser.add_option(
    "-r",
    "--remote",
    action="store",
    dest="remoterepo",
    default="codecommit",
    help="The name of the remote repository to be used as the push or migration
destination. The remote must already be set in the local repo ('git remote add ...').
If this option is not specified, the script will use 'codecommit' by default.",
)
parser.add_option(
    "-b",
    "--batch",
    action="store",
    dest="batchsize",
    default="1000",
    help="Specifies the commit batch size for pushes. If not explicitly set, the
default is 1,000 commits.",
)
parser.add_option(
    "-c",
    "--clean",
    action="store_true",
    dest="clean",
    default=False,
    help="Remove the temporary tags created by migration from both the local repo
and the remote repository. This option will not do any migration work, just cleanup.
Cleanup is done automatically at the end of a successful migration, but not after a
failure so that when you re-run the script, the tags from the prior run can be used to
identify commit batches that were not pushed successfully.",
)

(options, args) = parser.parse_args()

migration = RepositoryMigration()
migration.migrate_repository_in_parts(
    options.localrepo, options.remoterepo, int(options.batchsize), options.clean
)

```



Seguridad en AWS CodeCommit

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y un centro de datos que están diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre 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 los servicios de AWS en la nube de AWS. AWS también 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 conformidad de](#) . Para obtener información sobre los programas de conformidad que se aplican a AWS CodeCommit, consulte [Servicios de AWS en el ámbito del programa de conformidad](#) .
- Seguridad en la nube: su responsabilidad se determina según el servicio de AWS que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza CodeCommit. En los siguientes temas, se le mostrará cómo configurar CodeCommit para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que lo ayuden a monitorear y proteger los recursos de CodeCommit.

Temas

- [Protección de los datos en AWS CodeCommit](#)
- [Identity and Access Management para AWS CodeCommit](#)
- [Resiliencia en AWS CodeCommit](#)
- [Seguridad de la infraestructura en \(\) AWS CodeCommit](#)

Protección de los datos en AWS CodeCommit

Como se trata de un servicio administrado, está protegido por la seguridad de red global de AWS. 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.

Puede utilizar llamadas a la API publicadas en AWS para acceder 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.

Los repositorios de CodeCommit se cifran automáticamente en reposo. No se requiere ninguna acción por parte del cliente. CodeCommit también cifra los datos en tránsito del repositorio. Puede usar el protocolo HTTPS, el protocolo SSH o ambos con los repositorios de CodeCommit. Para obtener más información, consulte [Configuración de AWS CodeCommit](#). También puede configurar el [acceso entre cuentas](#) a los repositorios de CodeCommit.

Temas

- [AWS Key Management Service y cifrado para AWS CodeCommit repositorios](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

AWS Key Management Service y cifrado para AWS CodeCommit repositorios

Los datos de los CodeCommit repositorios se cifran en tránsito y en reposo. Cuando los datos se introducen en un CodeCommit repositorio (por ejemplo, mediante una llamada git push), CodeCommit cifra los datos recibidos tal como se almacenan en el repositorio. Cuando se extraen datos de un CodeCommit repositorio (por ejemplo, mediante una llamada git pull), CodeCommit los descifra y, a continuación, los envía a la persona que llama. Esto supone que el usuario de IAM asociado a la solicitud de inserción o extracción ha sido autenticado por AWS. Los datos enviados o recibidos se transmiten mediante los protocolos de red cifrada HTTPS o SSH.

Puede utilizar una clave gestionada por el cliente Clave administrada de AWS o una clave gestionada por el cliente para cifrar y descifrar los datos de su repositorio. Para obtener más información sobre las diferencias entre las claves administradas por el cliente y Claves administradas por AWS, consulte [Claves administradas por el cliente y Claves administradas por AWS](#). Si no

especificas una clave gestionada por el cliente, CodeCommit utilizará una Clave administrada de AWS para cifrar y descifrar los datos de tu repositorio. Este Clave administrada de AWS se crea automáticamente para usted en su Cuenta de AWS. La primera vez que crees un CodeCommit repositorio en una nueva Región de AWS cuenta de Amazon Web Services, si no especificas una clave gestionada por el cliente, CodeCommit crea una Clave administrada de AWS (la `aws/codecommit` clave) en esa misma cuenta Región de AWS en AWS Key Management Service (AWS KMS). Esta `aws/codecommit` clave solo la utilizan CodeCommit. Se almacena en su cuenta de Amazon Web Services. Según lo que especifique, utilizará CodeCommit la clave gestionada por el cliente o la Clave administrada de AWS para cifrar y descifrar los datos del repositorio.

 Important

CodeCommit realiza las siguientes AWS KMS acciones con la AWS KMS clave utilizada para cifrar y descifrar los datos de un repositorio. Si utilizas una Clave administrada de AWS, el usuario no necesita permisos explícitos para realizar estas acciones, pero no debe tener ninguna política adjunta que deniegue estas acciones para la `aws/codecommit` clave. Si utilizas una clave gestionada por el cliente que tiene un Cuenta de AWS identificador establecido como principal de política para esa clave, estos permisos deben estar configurados de forma explícita en `allow`. En concreto, cuando crees tu primer repositorio y si actualizas las claves de tu repositorio, no debes tener ninguno de los siguientes permisos establecidos `deny` si utilizas una Clave administrada de AWS, y debes configurarlos `allow` si utilizas una clave gestionada por el cliente con un principio de política:

- `"kms:Encrypt"`
- `"kms:Decrypt"`
- `"kms:ReEncrypt"` (según el contexto, esto podría requerir `kms:ReEncryptFrom`, `kms:ReEncryptTo`, o `kms:ReEncrypt*` no configurado en denegación)
- `"kms:GenerateDataKey"`
- `"kms:GenerateDataKeyWithoutPlaintext"`
- `"kms:DescribeKey"`

Si desea utilizar su propia clave gestionada por el cliente, la clave debe estar disponible en el Región de AWS lugar donde se encuentra el repositorio. CodeCommit admite el uso de claves administradas por clientes de una o varias regiones. Si bien se admiten todos los tipos de origen de materiales clave, recomendamos utilizar la opción KMS predeterminada. Los clientes que utilizan la opción de

Almacén de claves externo pueden sufrir retrasos por parte del proveedor de la tienda. Además, CodeCommit tiene los siguientes requisitos para las claves gestionadas por el cliente:

- CodeCommit solo admite el uso de claves simétricas.
- El tipo de uso de la clave debe estar establecido en Cifrar y descifrar.

Para obtener más información sobre la creación de claves administradas por el cliente, consulte [Conceptos](#) y [Creación de claves](#).

Para ver información sobre la Clave administrada de AWS generado por CodeCommit, haga lo siguiente:

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación de servicio, seleccione Claves administradas por AWS. Asegúrese de haber iniciado sesión en el Región de AWS lugar donde desea revisar las claves.
4. En la lista de claves de cifrado, elija la Clave administrada de AWS con el alias aws/codecommit. Se muestra información básica sobre el Clave propiedad de AWS .

No puede cambiarlo ni eliminarlo Clave administrada de AWS.

Cómo se utilizan los algoritmos de cifrado para cifrar los datos del repositorio

CodeCommit utiliza dos enfoques diferentes para cifrar los datos. Los objetos de Git individuales de menos de 6 MB se cifran utilizando AES-GCM-256, que proporciona validación de integridad de los datos. Los objetos de entre 6 MB y un máximo de 2 GB para un único blob se cifran mediante AES-CBC-256. CodeCommit siempre valida el contexto de cifrado.

Contexto de cifrado

Cada servicio integrado AWS KMS especifica un contexto de cifrado para las operaciones de cifrado y descifrado. El contexto de cifrado consta de información autenticada adicional que usa AWS KMS para comprobar la integridad de los datos. Si se ha especificado para la operación de cifrado, también debe especificarse en la operación de descifrado. De lo contrario, se produce un error en el descifrado. CodeCommit utiliza el identificador del CodeCommit repositorio para el contexto

de cifrado. Puede usar el `get-repository` comando o la CodeCommit consola para buscar el ID del repositorio. Busque el ID del CodeCommit repositorio en AWS CloudTrail los registros para saber qué operaciones de cifrado se realizaron y qué clave se AWS KMS utilizó para cifrar o descifrar los datos del CodeCommit repositorio.

Para obtener más información al respecto AWS KMS, consulte la Guía para [AWS Key Management Service desarrolladores](#).

Conexión a AWS CodeCommit repositorios con credenciales rotativas

Puede conceder a los usuarios acceso a sus AWS CodeCommit repositorios sin necesidad de configurar los usuarios de IAM para ellos ni utilizar una clave de acceso y una clave secreta. Para asignar permisos a identidades federadas, debe crear un rol y definir permisos para este. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte [Crear un rol para un proveedor de identidad de terceros \(federación\)](#) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center . También puede configurar el acceso basado en roles para que los usuarios de IAM accedan a los CodeCommit repositorios en cuentas independientes de Amazon Web Services (una técnica conocida como acceso entre cuentas). Para obtener un tutorial de configuración del acceso entre cuentas en un repositorio, consulte [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#).

También puede configurar el acceso para los usuarios que desean o deben autenticarse a través de métodos como:

- Lenguaje de marcado para confirmaciones de seguridad (SAML)
- Autenticación multifactor (MFA)
- Federación
- Login with Amazon
- Amazon Cognito
- Facebook
- Google
- Proveedor de identidad compatible con OpenID Connect (OIDC)

 Note

La siguiente información se aplica únicamente al uso `git-remote-codecommit` o al asistente de AWS CLI credenciales para conectarse a los repositorios. CodeCommit Dado que el enfoque recomendado para el acceso temporal o federado CodeCommit es la configuración `git-remote-codecommit`, en este tema se proporcionan ejemplos del uso de esa utilidad. Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

No puedes usar credenciales SSH o Git ni HTTPS para conectarte a CodeCommit repositorios con credenciales de acceso rotativas o temporales.

No es necesario que realice estos pasos si se cumplen todos los requisitos siguientes:

- Ha iniciado sesión en una instancia de Amazon EC2.
- Está utilizando Git y HTTPS con el asistente de AWS CLI credenciales para conectarse desde la instancia de Amazon EC2 a los repositorios. CodeCommit
- La instancia de EC2 tiene asociado un perfil de instancia de IAM que contiene los permisos de acceso definidos en [Para conexiones HTTPS en Linux, macOS o Unix con el asistente de AWS CLI credenciales](#) o [Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI](#).
- Ha instalado y configurado la aplicación de ayudante de credenciales de Git en la instancia de EC2 tal como se describe en [Para conexiones HTTPS en Linux, macOS o Unix con el asistente de AWS CLI credenciales](#) o [Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI](#).

Las instancias Amazon EC2 que cumplen los requisitos anteriores ya están configuradas para comunicarle las credenciales de acceso temporal CodeCommit en su nombre.

 Note

Puede configurar y utilizar `git-remote-codecommit` en instancias de EC2.

Para conceder a los usuarios acceso temporal a sus CodeCommit repositorios, siga estos pasos.

Paso 1: completar los requisitos previos

Complete los pasos de configuración para proporcionar a un usuario acceso a sus CodeCommit repositorios mediante credenciales rotativas:

- Para obtener más información sobre el acceso entre cuentas, consulte [Tutorial: Delegación del acceso entre cuentas de Amazon Web Services mediante roles de IAM](#) y [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#).
- Para conocer el SAML y la federación, consulta Cómo [usar el sistema de autenticación de tu organización para conceder acceso a AWS los recursos](#) y [Acerca de la federación basada en AWS STS SAML 2.0](#).
- Para obtener más información sobre MFA, consulte [Uso de dispositivos con autenticación multifactor \(MFA\) con AWS](#) y [Creación de credenciales de seguridad temporales para permitir el acceso a los usuarios de IAM](#).
- [Para iniciar sesión con Amazon, Amazon Cognito, Facebook, Google o cualquier proveedor de identidad compatible con OIDC, consulte Acerca de Web Identity Federation. AWS STS](#)

Utilice la información [Autenticación y control de acceso para AWS CodeCommit](#) para especificar los CodeCommit permisos que quiere conceder al usuario.

Paso 2: Obtener el nombre de rol o las credenciales de acceso

Si desea que los usuarios accedan a los repositorios asumiendo un rol, proporcione a los usuarios el nombre de recurso de Amazon (ARN) de ese rol. De lo contrario, dependiendo de la forma en que configure el acceso, el usuario puede obtener credenciales rotativas de una de las siguientes maneras:

- Para acceder a varias cuentas, ejecuta el comando AWS CLI [assume-role](#) o llama a la API. AWS STS [AssumeRole](#)
- [En el caso de SAML, llama al AWS CLI assume-role-with-saml comando o a la API de SAML. AWS STS AssumeRoleWith](#)
- Para la federación, llama a AWS CLI [assume-role](#) o [get-federation-token](#) comandos o a o. AWS STS [AssumeRoleGetFederationToken](#) APIs
- Para MFA, llame al AWS CLI [get-session-token](#) comando o a la API. AWS STS [GetSessionToken](#)

- Para Login with Amazon, Amazon Cognito, Facebook, Google o cualquier proveedor de identidad compatible con OIDC, llama al comando AWS CLI [assume-role-with-web-identity](#) o a la API AWS STS [AssumeRoleWithWebIdentity](#)

Paso 3: Instale y configure la git-remote-codecommit AWS CLI

Debe configurar el equipo local para que utilice las credenciales de acceso instalando [git-remote-codecommit](#) y configurando un perfil en la AWS CLI.

1. Siga las instrucciones que se detallan en [Configuración](#) para configurar la AWS CLI. Utilice el comando `aws configure` para configurar uno o varios perfiles. Considere la posibilidad de crear un perfil con nombre para usarlo cuando se conecte a CodeCommit los repositorios mediante credenciales rotativas.
2. Puede asociar las credenciales al perfil AWS CLI nombrado del usuario de una de las siguientes maneras.
 - Si está asumiendo un rol al que acceder CodeCommit, configure un perfil con nombre con la información necesaria para asumir ese rol. Por ejemplo, si desea asumir una función nombrada *CodeCommitAccess* en la cuenta de Amazon Web Services 1111, puede configurar un perfil predeterminado para usarlo cuando trabaje con otros AWS recursos y un perfil con nombre para usarlo cuando asuma esa función. Los siguientes comandos crean un nombre de perfil con nombre asignado *CodeAccess* que asume un rol denominado *CodeCommitAccess*. El nombre de usuario *Maria_Garcia* está asociado a la sesión y el perfil predeterminado se establece como la fuente de sus AWS credenciales:

```
aws configure set role_arn arn:aws:iam::111111111111:role/CodeCommitAccess --  
profile CodeAccess  
aws configure set source_profile default --profile CodeAccess  
aws configure set role_session_name "Maria_Garcia" --profile CodeAccess
```

Si desea verificar los cambios, consulte o edite manualmente el archivo `~/.aws/config` (para Linux) o el archivo `%UserProfile%.aws\config` (para Windows) y revise la información bajo el perfil con nombre. Por ejemplo, el archivo podría tener un aspecto similar al siguiente:

```
[default]  
region = us-east-1
```

```
output = json

[profile CodeAccess]
source_profile = default
role_session_name = Maria_Garcia
role_arn = arn:aws:iam::111111111111:role/CodeCommitAccess
```

Después de configurar el perfil con nombre, puede clonar repositorios de CodeCommit con la utilidad `git-remote-codecommit` utilizando el perfil con nombre. Por ejemplo, para clonar un repositorio denominado *MyDemoRepo*:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

- Si utiliza la federación de identidades web y OpenID Connect (OIDC), configure un perfil con nombre que haga la llamada a la `AssumeRoleWithWebIdentity` API AWS Security Token Service (AWS STS) en su nombre para actualizar las credenciales temporales. Utilice el `aws configure set` comando o edite manualmente el `~/.aws/credentials` archivo (para Linux) o el `%UserProfile%.aws\credentials` archivo (para Windows) para añadir un perfil con AWS CLI nombre y los valores de configuración necesarios. Por ejemplo, para crear un perfil que asuma la *CodeCommitAccess* función y utilice un archivo de token de identidad web `~/my-credentials/my-token-file`

```
[CodeCommitWebIdentity]
role_arn = arn:aws:iam::111111111111:role/CodeCommitAccess
web_identity_token_file=~/my-credentials/my-token-file
role_session_name = Maria_Garcia
```

Para obtener más información, consulte [Configuración de AWS Command Line Interface](#) y [Uso de un rol de IAM en la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface .

Paso 4: Acceder a los repositorios CodeCommit

Suponiendo que el usuario haya seguido las instrucciones para conectarse [Conectarse a un repositorio](#) a CodeCommit los repositorios, utilizará la funcionalidad ampliada proporcionada por `git-remote-codecommit` Git para llamar `git clone` y `git pull` clonar, enviar y extraer datos de los CodeCommit repositorios a los que tiene acceso. `git push` Por ejemplo, para clonar un repositorio:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

Los comandos commit, push y pull de Git usan la sintaxis normal de Git.

Cuando el usuario usa AWS CLI y especifica el AWS CLI nombre del perfil asociado a las credenciales de acceso rotativas, se muestran los resultados relacionados con ese perfil.

Identity and Access Management para AWS CodeCommit

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. CodeCommit La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración del acceso con políticas](#)
- [Autenticación y control de acceso para AWS CodeCommit](#)
- [¿Cómo AWS CodeCommit funciona con IAM](#)
- [CodeCommit políticas basadas en recursos](#)
- [Autorización basada en etiquetas CodeCommit](#)
- [CodeCommit Funciones de IAM](#)
- [AWS CodeCommit ejemplos de políticas basadas en la identidad](#)
- [Solución de problemas AWS CodeCommit de identidad y acceso](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte [Solución de problemas AWS CodeCommit de identidad y acceso](#)).

- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [¿Cómo AWS CodeCommit funciona con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [AWS CodeCommit ejemplos de políticas basadas en la identidad](#)).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver la lista completa de las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe [especificar una entidad principal](#) en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la [descripción general de la lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos otorgados por los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

Autenticación y control de acceso para AWS CodeCommit

El acceso AWS CodeCommit requiere credenciales. Esas credenciales deben tener permisos para acceder a AWS los recursos, como los CodeCommit repositorios, y a tu usuario de IAM, que utilizas para gestionar tus credenciales de Git o la clave pública de SSH que utilizas para realizar las conexiones de Git. En las siguientes secciones se proporciona información detallada sobre cómo puedes usar [AWS Identity and Access Management \(IAM\)](#) y cómo ayudar CodeCommit a proteger el acceso a tus recursos:

- [Autenticación](#)
- [Control de acceso](#)

Autenticación

Como CodeCommit los repositorios están basados en Git y admiten las funciones básicas de Git, incluidas las credenciales de Git, te recomendamos que utilices un usuario de IAM cuando trabajes con ellos. CodeCommit Puedes acceder CodeCommit con otros tipos de identidad, pero los demás tipos de identidad están sujetos a limitaciones, como se describe a continuación.

Tipos de identidades:

- Usuario de IAM: un [usuario de IAM](#) es simplemente una identidad en su Amazon Web Services que tiene permisos personalizados específicos. Por ejemplo, un usuario de IAM puede tener permisos para crear y gestionar las credenciales de Git para acceder a los CodeCommit repositorios. Este es el tipo de usuario con el que se recomienda trabajar. CodeCommit [Puede utilizar un nombre de usuario y una contraseña de IAM para iniciar sesión en AWS páginas web seguras Consola de administración de AWS, como los foros de AWS debate o el AWS Support Centro.](#)

Puede generar credenciales de Git o asociar claves SSH públicas a su usuario de IAM o puede instalar y configurar git-remote-codecommit. Estas son las formas más sencillas de configurar Git para que funcione con tus CodeCommit repositorios. Con las [credenciales de Git](#), puede

generar un nombre de usuario y contraseña estáticos en IAM. Seguidamente, puede utilizar estas credenciales para las conexiones HTTPS con Git y cualquier herramienta de terceros que admita la autenticación con nombre de usuario y contraseña de Git. Con las conexiones SSH, creas archivos de claves públicas y privadas en tu máquina local que Git CodeCommit usa para la autenticación SSH. Asocie la clave pública con el usuario de IAM y almacene la clave privada en el equipo local. [git-remote-codecommit](#) amplía la funcionalidad de Git y no requiere la configuración de credenciales de Git para el usuario.

Además, puede generar [claves de acceso](#) para cada usuario. [Usa las claves de acceso cuando accedas a AWS los servicios mediante programación, ya sea a través de una de ellas AWS SDKs o mediante la tecla \(\)](#). [AWS Command Line Interface](#) [AWS CLI](#) El SDK y las herramientas de CLI usan claves de acceso para firmar criptográficamente las solicitudes. Si no usa las AWS herramientas, debe firmar las solicitudes usted mismo. CodeCommit es compatible con la versión 4 de Signature, un protocolo para autenticar las solicitudes de API entrantes. Para obtener más información acerca de la autenticación de solicitudes, consulte [Proceso de firma Signature Version 4](#) en la Referencia general de AWS.

- Usuario raíz de la cuenta de Amazon Web Services: al registrarse AWS, proporciona una dirección de correo electrónico y una contraseña asociadas a su cuenta de Amazon Web Services. Estas son las credenciales raíz y proporcionan acceso completo a todos los recursos de AWS . Algunas CodeCommit funciones no están disponibles para los usuarios de la cuenta raíz. Además, la única forma de usar Git con tu cuenta raíz es instalar y configurar git-remote-codecommit (recomendado) o configurar el asistente de AWS credenciales, que se incluye con la. AWS CLI No puede utilizar las credenciales de Git o pares de claves públicas y privadas SSH con su usuario de la cuenta raíz. Por estos motivos, no recomendamos utilizar el usuario de la cuenta raíz cuando interactúes con él. CodeCommit

Important

Por motivos de seguridad, le recomendamos que utilice las credenciales raíz solo para crear un usuario administrador, que es un usuario de IAM con permiso total para administrar su cuenta de AWS . A continuación, puede utilizar este usuario administrador para crear otros usuarios y roles de IAM; con permisos limitados. Para obtener más información, consulte [Prácticas recomendadas de IAM](#) y [Creación de un grupo y usuario administrador](#) en la Guía del usuario de IAM.

- IAM Identity Center y los usuarios del IAM Identity Center: AWS IAM Identity Center amplían las capacidades AWS Identity and Access Management para proporcionar un lugar central que reúna

la administración de los usuarios y su acceso a Cuentas de AWS las aplicaciones en la nube. Si bien es una práctica recomendada para la mayoría de los usuarios que trabajan con AWS, IAM Identity Center actualmente no proporciona mecanismos para las credenciales de Git o los pares de claves SSH. Estos usuarios pueden instalarlos y configurarlos git-remote-codecommit para clonar CodeCommit repositorios de forma local, pero no todos los entornos de desarrollo integrados (IDEs) admiten la clonación, la inserción o la extracción. git-remote-codecommit

Como práctica recomendada, exija a los usuarios humanos que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

- Rol de IAM: al igual que un usuario de IAM, un [rol de IAM](#) es una identidad de IAM que puede crear en la cuenta para conceder permisos específicos.

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de API. AWS Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Note

No puede utilizar las credenciales de Git o pares de claves públicas y privadas SSH con usuarios federados. Además, las preferencias del usuario no están disponibles para los usuarios federados. Para obtener información acerca de cómo configurar conexiones mediante el acceso federado, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Control de acceso

Puede tener credenciales válidas para autenticar sus solicitudes, pero a menos que tenga permisos, no podrá crear CodeCommit recursos ni acceder a ellos. Por ejemplo, debe disponer de permisos para ver repositorios, enviar código, crear y administrar credenciales de Git, etc.

En las siguientes secciones se describe cómo administrar los permisos para CodeCommit. Recomendamos que lea primero la información general.

- [Descripción general de la administración de los permisos de acceso a sus CodeCommit recursos](#)
- [Uso de políticas basadas en la identidad \(políticas de IAM\) para CodeCommit](#)
- [CodeCommit referencia de permisos](#)

Descripción general de la administración de los permisos de acceso a sus CodeCommit recursos

Cada AWS recurso pertenece a una cuenta de Amazon Web Services. Los permisos de creación o acceso a un recurso se rigen por políticas de permisos. Un administrador de cuentas puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y funciones). Algunos servicios, por ejemplo AWS Lambda, también permiten adjuntar políticas de permisos a los recursos.

Note

Un administrador de cuentas (o usuario administrador) es un usuario que tiene privilegios de administrador. Para obtener más información, consulte [Prácticas recomendadas de IAM](#) en la Guía del usuario de IAM.

Al conceder permisos, usted decide quién obtiene los permisos, para qué recursos se obtienen permisos y qué acciones específicas desea permitir en esos recursos.

Temas

- [CodeCommit recursos y operaciones](#)
- [Titularidad de los recursos](#)
- [Administración del acceso a los recursos](#)
- [El alcance de los recursos en CodeCommit](#)

- [Especificar elementos de políticas: recursos, acciones, efectos y entidades principales](#)
- [Especificación de las condiciones de una política](#)

CodeCommit recursos y operaciones

En CodeCommit, el recurso principal es un repositorio. Cada recurso tiene asociado un nombre de recurso de Amazon (ARN) único. En las políticas se emplean nombres de recurso de Amazon (ARN) para identificar los recursos a los que se aplican las políticas. Para obtener más información ARNs, consulte [Amazon Resource Names \(ARN\) y AWS Service Namespaces](#) en. Referencia general de Amazon Web Services CodeCommit actualmente no admite otros tipos de recursos, que se denominan subrecursos.

En la siguiente tabla se describe cómo especificar CodeCommit los recursos.

Tipo de recurso	Formato de ARN
Repository	arn:aws:codecommit::: <i>region account-id repository-name</i>
Todos los CodeCommit repositorios	arn:aws:codecommit:*
Todos los CodeCommit repositorios pertenecen a la cuenta especificada en la especificada Región de AWS	arn:aws:codecommit::: * <i>region account-id</i>

Note

La mayoría de AWS los servicios utilizan dos puntos (:) o una barra diagonal (/) como el mismo carácter. ARNs Sin embargo, CodeCommit requiere una coincidencia exacta en los patrones y reglas de los recursos. Al crear patrones de eventos, asegúrese de utilizar los caracteres de ARN correctos para que coincidan con la sintaxis de ARN en el recurso.

Por ejemplo, puede indicar un repositorio específico (*MyDemoRepo*) en la instrucción usando su ARN de este modo:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo"
```

Para especificar todos los repositorios que pertenecen a una cuenta específica, use el carácter comodín (*) del modo siguiente:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:*"
```

Para especificar todos los recursos, o si una acción específica de la API no es compatible ARNs, utiliza el carácter comodín (*) en el Resource elemento de la siguiente manera:

```
"Resource": "*"
```

También puede utilizar el carácter comodín (*) para especificar todos los recursos que coinciden con parte de un nombre de repositorio. Por ejemplo, el siguiente ARN especifica cualquier CodeCommit repositorio que comience por el nombre MyDemo y que esté registrado en la cuenta de Amazon Web Services 111111111111 en: us-east-2 Región de AWS

```
arn:aws:codecommit:us-east-2:111111111111:MyDemo*
```

Para obtener una lista de las operaciones disponibles que funcionan con los CodeCommit recursos, consulte [CodeCommit referencia de permisos](#).

Titularidad de los recursos

La cuenta de Amazon Web Services es la propietaria de los recursos que se crean en la cuenta, independientemente de quién los haya creado. En concreto, el propietario de los recursos es la cuenta de Amazon Web Services es la [entidad principal](#) (es decir, la cuenta raíz, un usuario de IAM o un rol de IAM) que autentica la solicitud de creación de recursos. Los siguientes ejemplos ilustran cómo funciona:

- Si crea un usuario de IAM en su cuenta de Amazon Web Services y concede permisos para crear CodeCommit recursos a ese usuario, el usuario podrá crear CodeCommit recursos. Sin embargo, su cuenta de Amazon Web Services, a la que pertenece el usuario, es propietaria de los CodeCommit recursos.
- Si utiliza las credenciales de la cuenta raíz de su cuenta de Amazon Web Services para crear una regla, su cuenta de Amazon Web Services es la propietaria del CodeCommit recurso.
- Si crea un rol de IAM en su cuenta de Amazon Web Services con permisos para crear CodeCommit recursos, cualquier persona que pueda asumir el rol podrá crear CodeCommit

recursos. Su cuenta de Amazon Web Services, a la que pertenece el rol, es propietaria de los CodeCommit recursos.

Administración del acceso a los recursos

Para administrar el acceso a AWS los recursos, se utilizan políticas de permisos. Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones para crear políticas de permisos.

Note

En esta sección se analiza el uso de la IAM en el contexto de CodeCommit. No se proporciona información detallada sobre el servicio de IAM. Para obtener más información acerca de IAM, consulte [¿Qué es IAM?](#) en la Guía del usuario de IAM. Para obtener más información acerca de la sintaxis y las descripciones de las políticas del IAM, consulte [Referencia de políticas de IAM de](#) en la Guía del usuario de IAM.

Las políticas de permisos que se asocian a una identidad de IAM se denominan políticas basadas en la identidad (políticas de IAM). Las políticas de permisos que se asocian a un recurso se denominan políticas basadas en recursos. Actualmente, solo CodeCommit admite políticas basadas en la identidad (políticas de IAM).

Temas

- [Políticas basadas en identidades \(políticas de IAM\)](#)
- [Políticas basadas en recursos](#)

Políticas basadas en identidades (políticas de IAM)

Para administrar el acceso a AWS los recursos, debe adjuntar políticas de permisos a las identidades de IAM. En CodeCommit, se utilizan políticas basadas en la identidad para controlar el acceso a los repositorios. Por ejemplo, puede hacer lo siguiente:

- Adjunta una política de permisos a un usuario o grupo de tu cuenta: para conceder a un usuario permisos para ver los CodeCommit recursos de la CodeCommit consola, adjunta una política de permisos basada en la identidad al usuario o grupo al que pertenezca el usuario.

- Asociar una política de permisos a un rol (para conceder permisos entre cuentas): la delegación como, por ejemplo, cuando se desea conceder acceso entre cuentas, implica el establecimiento de una relación de confianza entre la cuenta que posee el recurso (la cuenta que confía) y la cuenta que contiene los usuarios que necesitan acceder al recurso (la cuenta en la que se confía). Una política de permisos concede al usuario del rol los permisos necesarios para realizar las tareas previstas en el recurso. Una política de confianza especifica las cuentas de confianza que pueden conceder permisos a sus usuarios para asumir el rol. Para obtener más información, consulte [Términos y conceptos sobre IAM](#).

Para conceder permisos entre cuentas, asocie una política de permisos basada en identidad a un rol de IAM. Por ejemplo, el administrador de la cuenta A puede crear un rol para conceder permisos multicuenta a otra cuenta de Amazon Web Services (por ejemplo, la cuenta B) o a un AWS servicio de la siguiente manera:

1. El administrador de la CuentaA crea un rol de IAM y asocia una política de permisos a dicho rol, que concede permisos sobre los recursos de la CuentaA.
2. El administrador de la CuentaA asocia una política de confianza al rol que identifica la Cuenta B como la entidad principal que puede asumir el rol.
3. De este modo, el administrador de la cuenta B puede delegar los permisos para que asuman el rol en cualquier usuario de la cuenta B. De este modo, los usuarios de la cuenta B pueden crear recursos de la cuenta A. Si desea conceder un permiso de AWS servicio para que asuma el rol, el principal de la política de confianza también puede ser el principal de AWS servicio. Para obtener más información, consulte Delegación en [Términos y conceptos sobre IAM](#).

Para obtener más información sobre el uso de IAM para delegar permisos, consulte [Administración de accesos](#) en la Guía del usuario de IAM.

La siguiente política de ejemplo permite a un usuario crear una ramificación en un repositorio denominado *MyDemoRepo*:

JSON

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

```
"Action" : [
    "codecommit:CreateBranch"
],
"Resource" : "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
}
]
}
```

Para restringir las llamadas y los recursos a los que los usuarios de su cuenta tienen acceso, cree políticas específicas de IAM y, a continuación, asocie esas políticas a usuarios de IAM. Para obtener más información sobre cómo crear funciones de IAM y explorar ejemplos de declaraciones de política de IAM CodeCommit, consulte [Ejemplos de políticas de identidad administradas por los clientes](#)

Políticas basadas en recursos

Algunos servicios, como Amazon S3, también admiten políticas de permisos basadas en recursos. Por ejemplo, puede adjuntar una política basada en recursos a un bucket de S3 para gestionar los permisos de acceso a ese bucket. CodeCommit no admite políticas basadas en recursos, pero puedes usar etiquetas para identificar los recursos, que luego puedes usar en las políticas de IAM. Para ver un ejemplo de una política basada en etiquetas, consulte [Políticas basadas en identidades \(políticas de IAM\)](#).

El alcance de los recursos en CodeCommit

En CodeCommit, puede establecer el alcance de las políticas y los permisos basados en la identidad para los recursos, tal y como se describe en [CodeCommit recursos y operaciones](#). Sin embargo, no puede definir el ámbito del permiso `ListRepositories` para un recurso. En su lugar, debe definir el ámbito para todos los recursos (utilizando el comodín `*`). De no hacerlo, la acción dará error.

Todos los demás CodeCommit permisos se pueden limitar a los recursos.

Especificar elementos de políticas: recursos, acciones, efectos y entidades principales

Puede crear políticas para permitir o denegar a los usuarios el acceso a los recursos, o permitir o denegar a los usuarios que realicen acciones específicas en esos recursos. CodeCommit define un conjunto de operaciones de API públicas que definen la forma en que los usuarios trabajan con el servicio, ya sea a través de la CodeCommit consola SDKs AWS CLI, la consola o llamándolas directamente APIs. Para conceder permisos para estas operaciones de la API, CodeCommit define un conjunto de acciones que puedes especificar en una política.

Algunas operaciones API pueden requerir permisos para realizar más de una acción. Para obtener más información sobre los recursos y las operaciones de API, consulte [CodeCommit recursos y operaciones](#) y [CodeCommit referencia de permisos](#).

A continuación se indican los elementos básicos de una política:

- **Recurso:** para identificar el recurso al que se aplica la política, use un nombre de recurso de Amazon (ARN). Para obtener más información, consulte [CodeCommit recursos y operaciones](#).
- **Acción:** para identificar las operaciones del recurso que desea permitir o denegar, utilice palabras de clave de acción. Por ejemplo, según lo especificado `Effect`, el `codecommit:GetBranch` permiso permite o deniega al usuario realizar la `GetBranch` operación, lo que obtiene detalles sobre una rama de un CodeCommit repositorio.
- **Efecto:** especifique el efecto que se producirá cuando el usuario solicite la acción específica, ya sea permitir o denegar. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso para asegurarse de que un usuario no pueda obtener acceso a él, aunque otra política le conceda acceso.
- **Principal:** en las políticas basadas en la identidad (políticas de IAM), el único tipo de políticas que CodeCommit admite, es el usuario al que está vinculada la política, es el principal implícito.

Para obtener más información sobre la sintaxis de las políticas de IAM consulte la [Referencia sobre la política de IAM](#) en la Guía del usuario de IAM.

Para ver una tabla que muestra todas las acciones de la CodeCommit API y los recursos a los que se aplican, consulte. [CodeCommit referencia de permisos](#)

Especificación de las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de acceso de IAM para especificar las condiciones en las que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte [Condición](#) y [Gramática de políticas](#) en la Guía de usuario de IAM.

Cómo expresar condiciones, se usan claves de condición predefinidas. No hay claves de condición específicas para CodeCommit. Sin embargo, hay claves AWS de condición generales que puede utilizar según convenga. Para obtener una lista completa de las claves AWS de ancho, consulte las [claves disponibles para las condiciones](#) en la Guía del usuario de IAM.

Uso de políticas basadas en la identidad (políticas de IAM) para CodeCommit

Los siguientes ejemplos de políticas basadas en la identidad demuestran cómo un administrador de cuentas puede adjuntar políticas de permisos a las identidades de IAM (usuarios, grupos y funciones) a fin de conceder permisos para realizar operaciones en los recursos. CodeCommit

Important

Le recomendamos que consulte primero los temas introductorios en los que se explican los conceptos básicos y las opciones disponibles para administrar el acceso a sus recursos. CodeCommit Para obtener más información, consulte [Descripción general de la administración de los permisos de acceso a sus CodeCommit recursos](#).

Temas

- [Permisos necesarios para usar la consola CodeCommit](#)
- [Visualización de recursos en la consola](#)
- [AWS políticas gestionadas para CodeCommit](#)
- [Ejemplos de políticas administradas por el cliente](#)

A continuación, se muestra un ejemplo de una política de permisos basada en identidad:

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codecommit:BatchGetRepositories"
      ],
      "Resource" : [
        "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
        "arn:aws:codecommit:us-east-2:111122223333:MyDemo*"
      ]
    }
  ]
}
```

```
}
```

Esta política tiene una declaración que permite al usuario obtener información sobre el CodeCommit repositorio nombrado `MyDestinationRepo` y todos los CodeCommit repositorios que comienzan con ese nombre `MyDemo` en la **us-east-2** región.

Permisos necesarios para usar la consola CodeCommit

Para ver los permisos necesarios para cada operación de la CodeCommit API y obtener más información sobre CodeCommit las operaciones, consulte [CodeCommit referencia de permisos](#).

Para permitir que los usuarios usen la CodeCommit consola, el administrador debe concederles permisos para CodeCommit realizar acciones. Por ejemplo, puede adjuntar la política [AWSCodeCommitPowerUser](#) administrada o su equivalente a un usuario o grupo.

Además de los permisos que otorgan a los usuarios las políticas basadas en la identidad, CodeCommit requiere permisos para las acciones de AWS Key Management Service (AWS KMS). Un usuario de IAM no necesita permisos explícitos de `Allow` para estas acciones, pero el usuario no debe tener políticas asociadas que definan los siguientes permisos como `Deny`:

```
"kms:Encrypt",
"kms:Decrypt",
"kms:ReEncrypt",
"kms:GenerateDataKey",
"kms:GenerateDataKeyWithoutPlaintext",
"kms:DescribeKey"
```

Para obtener más información sobre el cifrado CodeCommit, consulte. [AWS KMS y cifrado](#)

Visualización de recursos en la consola

La CodeCommit consola necesita el `ListRepositories` permiso para mostrar una lista de los repositorios de su cuenta de Amazon Web Services en la Región de AWS que ha iniciado sesión. La consola también incluye una función `Go to resource` (Ir al recurso) para realizar una búsqueda rápida de recursos sin distinción entre mayúsculas y minúsculas. Esta búsqueda se realiza en su cuenta de Amazon Web Services en la Región de AWS que ha iniciado sesión. Los siguientes recursos se muestran en los siguientes servicios:

- AWS CodeBuild: proyectos de compilación

- AWS CodeCommit: repositorios
- AWS CodeDeploy: aplicaciones
- AWS CodePipeline: canalizaciones

Para realizar esta búsqueda en los recursos de todos los servicios, debe contar con los siguientes permisos:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Los resultados de los recursos de un servicio no se devuelven si no tiene permisos para ese servicio. Aunque tenga permisos para ver los recursos, no se devolverán recursos específicos si hay un permiso Deny explícito para ver esos recursos.

AWS políticas gestionadas para CodeCommit

Para añadir permisos a usuarios, grupos y roles, es más fácil usar políticas AWS administradas que escribirlas usted mismo. Se necesita tiempo y experiencia para [crear políticas administradas por el cliente de IAM](#) que proporcionen a su equipo solo los permisos necesarios. Para empezar rápidamente, puedes usar nuestras políticas AWS gestionadas. Estas políticas cubren casos de uso comunes y están disponibles en su Cuenta de AWS. Para obtener más información sobre las políticas AWS administradas, consulte las [políticas AWS administradas](#) en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas. No puede cambiar los permisos en las políticas AWS gestionadas. En ocasiones, los servicios agregan permisos adicionales a una política administrada de AWS para admitir características nuevas. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Es más probable que los servicios actualicen una política administrada de AWS cuando se lanza una nueva característica o cuando se ponen a disposición nuevas operaciones. Los servicios no eliminan los permisos de una política AWS administrada, por lo que las actualizaciones de la política no afectarán a los permisos existentes.

Además, AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política `ReadOnlyAccess` AWS gestionada proporciona acceso de solo lectura a

todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte [Políticas administradas de AWS para funciones de trabajo](#) en la Guía del usuario de IAM.

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por AWS. Estas políticas AWS gestionadas otorgan los permisos necesarios para casos de uso comunes. Las políticas gestionadas CodeCommit también proporcionan permisos para realizar operaciones en otros servicios, como IAM, Amazon SNS y CloudWatch Amazon Events, según lo exijan las responsabilidades de los usuarios a los que se ha concedido la política en cuestión. Por ejemplo, la `AWSCodeCommitFullAccess` política es una política de usuario de nivel administrativo que permite a los usuarios con esta política crear y gestionar reglas de CloudWatch eventos para los repositorios (reglas cuyos nombres llevan el prefijo) y temas de codecommit Amazon SNS para notificaciones sobre eventos relacionados con el repositorio (temas cuyos nombres llevan el prefijo), así como administrar los repositorios en codecommit CodeCommit.

Las siguientes políticas AWS administradas, que puede adjuntar a los usuarios de su cuenta, son específicas de cada una. CodeCommit

Temas

- [AWS política gestionada: AWSCodeCommitFullAccess](#)
- [AWS política gestionada: AWSCodeCommitPowerUser](#)
- [AWS política gestionada: AWSCodeCommitReadOnly](#)
- [CodeCommit políticas y notificaciones gestionadas](#)
- [AWS CodeCommit políticas gestionadas y Amazon CodeGuru Reviewer](#)
- [CodeCommit actualizaciones de las políticas gestionadas AWS](#)

AWS política gestionada: AWSCodeCommitFullAccess

Puede asociar la política `AWSCodeCommitFullAccess` a las identidades de IAM. Esta política otorga acceso completo a CodeCommit. Aplique esta política únicamente a los usuarios de nivel administrativo a los que desee conceder un control total sobre los CodeCommit repositorios y los recursos relacionados de su cuenta de Amazon Web Services, incluida la posibilidad de eliminar repositorios.

La `AWSCodeCommitFullAccess` política contiene la siguiente declaración de política:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchEventsCodeCommitRulesAccess",
      "Effect": "Allow",
      "Action": [
        "events:DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:ListTargetsByRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/codecommit*"
    },
    {
      "Sid": "SNSTopicAndSubscriptionAccess",
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:Subscribe",
        "sns:Unsubscribe",
        "sns:SetTopicAttributes"
      ],
      "Resource": "arn:aws:sns:*:*:codecommit*"
    },
    {
      "Sid": "SNSTopicAndSubscriptionReadAccess",
      "Effect": "Allow",
      "Action": [
```

```

        "sns:ListTopics",
        "sns:ListSubscriptionsByTopic",
        "sns:GetTopicAttributes"
    ],
    "Resource": "*"
},
{
    "Sid": "LambdaReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
        "lambda:ListFunctions"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListUsers"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListAccessKeys",
        "iam:ListSSHPublicKeys",
        "iam:ListServiceSpecificCredentials"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "IAMUserSSHKeys",
    "Effect": "Allow",
    "Action": [
        "iam:DeleteSSHPublicKey",
        "iam:GetSSHPublicKey",
        "iam:ListSSHPublicKeys",
        "iam:UpdateSSHPublicKey",
        "iam:UploadSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},

```

```

{
  "Sid": "IAMSelfManageServiceSpecificCredentials",
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceSpecificCredential",
    "iam:UpdateServiceSpecificCredential",
    "iam:DeleteServiceSpecificCredential",
    "iam:ResetServiceSpecificCredential"
  ],
  "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications>DeleteNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "codestar-notifications:NotificationsForResource":
"arn:aws:iam::*:role/Service*"
    }
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
  "Effect": "Allow",

```



```

    "Action": [
        "sns:CreateTopic",
        "sns:SetTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
    "Sid": "AmazonCodeGuruReviewerFullAccess",
    "Effect": "Allow",
    "Action": [
        "codeguru-reviewer:AssociateRepository",
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DisassociateRepository",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerSLRCreation",
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam:*:*:role/aws-service-role/codeguru-
reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
        }
    }
}

```

```

    }
  },
  {
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeStarConnectionsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-connections:ListConnections",
      "codestar-connections:GetConnection"
    ],
    "Resource": "arn:aws:codestar-connections:*:*:connection/*"
  }
]
}

```

AWS política gestionada: AWSCode CommitPowerUser

Puede asociar la política AWSCodeCommitPowerUser a las identidades de IAM. Esta política permite a los usuarios acceder a todas las funciones CodeCommit y los recursos relacionados con el repositorio, excepto que no les permite eliminar CodeCommit repositorios ni crear o eliminar recursos relacionados con el repositorio en otros servicios AWS , como Amazon Events. CloudWatch Le recomendamos que aplique esta política a la mayoría de los usuarios.

La AWSCode CommitPowerUser política contiene la siguiente declaración de política:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",

```

```

    "Action": [
      "codecommit:AssociateApprovalRuleTemplateWithRepository",
      "codecommit:BatchAssociateApprovalRuleTemplateWithRepositories",
      "codecommit:BatchDisassociateApprovalRuleTemplateFromRepositories",
      "codecommit:BatchGet*",
      "codecommit:BatchDescribe*",
      "codecommit:Create*",
      "codecommit>DeleteBranch",
      "codecommit>DeleteFile",
      "codecommit:Describe*",
      "codecommit:DisassociateApprovalRuleTemplateFromRepository",
      "codecommit:EvaluatePullRequestApprovalRules",
      "codecommit:Get*",
      "codecommit:List*",
      "codecommit:Merge*",
      "codecommit:OverridePullRequestApprovalRules",
      "codecommit:Put*",
      "codecommit:Post*",
      "codecommit:TagResource",
      "codecommit:Test*",
      "codecommit:UntagResource",
      "codecommit:Update*",
      "codecommit:GitPull",
      "codecommit:GitPush"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CloudWatchEventsCodeCommitRulesAccess",
    "Effect": "Allow",
    "Action": [
      "events:DeleteRule",
      "events:DescribeRule",
      "events:DisableRule",
      "events:EnableRule",
      "events:PutRule",
      "events:PutTargets",
      "events:RemoveTargets",
      "events:ListTargetsByRule"
    ],
    "Resource": "arn:aws:events:*:*:rule/codecommit*"
  },
  {
    "Sid": "SNSTopicAndSubscriptionAccess",

```

```
    "Effect": "Allow",
    "Action": [
        "sns:Subscribe",
        "sns:Unsubscribe"
    ],
    "Resource": "arn:aws:sns:*:*:codecommit*"
},
{
    "Sid": "SNSTopicAndSubscriptionReadAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics",
        "sns:ListSubscriptionsByTopic",
        "sns:GetTopicAttributes"
    ],
    "Resource": "*"
},
{
    "Sid": "LambdaReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
        "lambda:ListFunctions"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListUsers"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListAccessKeys",
        "iam:ListSSHPublicKeys",
        "iam:ListServiceSpecificCredentials"
    ],
    "Resource": "arn:aws:iam:*:*:user/${aws:username}"
},
{
```

```

    "Sid": "IAMUserSSHKeys",
    "Effect": "Allow",
    "Action": [
        "iam:DeleteSSHPublicKey",
        "iam:GetSSHPublicKey",
        "iam:ListSSHPublicKeys",
        "iam:UpdateSSHPublicKey",
        "iam:UploadSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "IAMSelfManageServiceSpecificCredentials",
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceSpecificCredential",
        "iam:UpdateServiceSpecificCredential",
        "iam:DeleteServiceSpecificCredential",
        "iam:ResetServiceSpecificCredential"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "codestar-notifications:NotificationsForResource":
"arn:aws:iam::*:role/Service"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [

```

```

        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerFullAccess",
    "Effect": "Allow",
    "Action": [
        "codeguru-reviewer:AssociateRepository",
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DisassociateRepository",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerSLRCreation",
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/codeguru-
reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {

```

```

        "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
    }
}
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarConnectionsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
    ],
    "Resource": "arn:aws:codestar-connections:*:*:connection/*"
}
]
}

```

AWS política gestionada: AWSCode CommitReadOnly

Puede asociar la política AWSCodeCommitReadOnly a las identidades de IAM. Esta política otorga acceso de solo lectura a CodeCommit los recursos relacionados con el repositorio en otros AWS servicios, así como la capacidad de crear y administrar sus propios recursos CodeCommit relacionados (como las credenciales de Git y las claves SSH para que los usuarios de IAM las utilicen al acceder a los repositorios). Aplique esta política a los usuarios a los que desea conceder la capacidad de leer el contenido de un repositorio, pero no realizar ningún cambio en su contenido.

La política contiene la siguiente declaración de política: AWSCode CommitReadOnly

JSON

```

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

```

```
{
  "Effect": "Allow",
  "Action": [
    "codecommit:BatchGet*",
    "codecommit:BatchDescribe*",
    "codecommit:Describe*",
    "codecommit:EvaluatePullRequestApprovalRules",
    "codecommit:Get*",
    "codecommit:List*",
    "codecommit:GitPull"
  ],
  "Resource": "*"
},
{
  "Sid": "CloudWatchEventsCodeCommitRulesReadOnlyAccess",
  "Effect": "Allow",
  "Action": [
    "events:DescribeRule",
    "events:ListTargetsByRule"
  ],
  "Resource": "arn:aws:events:*:*:rule/codecommit*"
},
{
  "Sid": "SNSSubscriptionAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics",
    "sns:ListSubscriptionsByTopic",
    "sns:GetTopicAttributes"
  ],
  "Resource": "*"
},
{
  "Sid": "LambdaReadOnlyListAccess",
  "Effect": "Allow",
  "Action": [
    "lambda:ListFunctions"
  ],
  "Resource": "*"
},
{
  "Sid": "IAMReadOnlyListAccess",
  "Effect": "Allow",
  "Action": [
```



```

        "iam:ListUsers"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListAccessKeys",
        "iam:ListSSHPublicKeys",
        "iam:ListServiceSpecificCredentials",
        "iam:GetSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "CodeStarNotificationsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:DescribeNotificationRule"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "codestar-
notifications:NotificationsForResource": "arn:aws:codecommit:us-
east-2:111122223333:*"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerReadOnlyAccess",
    "Effect": "Allow",
    "Action": [

```

```

        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarConnectionsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
    ],
    "Resource": "arn:aws:codestar-connections:*:*:connection/*"
}
]
}

```

CodeCommit políticas y notificaciones gestionadas

AWS CodeCommit admite las notificaciones, que pueden notificar a los usuarios los cambios importantes en los repositorios. Las políticas gestionadas CodeCommit incluyen declaraciones de políticas para la funcionalidad de notificación. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

Permisos relacionados con las notificaciones en políticas administradas de acceso total

La política administrada `AWSCodeCommitFullAccess` incluye las siguientes instrucciones para permitir el acceso completo a las notificaciones. Los usuarios con esta política administrada aplicada también pueden crear y administrar temas de Amazon SNS para notificaciones, suscribirse y cancelar la suscripción de los usuarios a los temas, mostrar los temas que se pueden elegir como destinos para las reglas de notificación y mostrar Amazon Q Developer en los clientes de aplicaciones de chat configurados para Slack.

```

{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",

```

```

        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codecommit:*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
    "Effect": "Allow",
    "Action": [
        "sns:CreateTopic",
        "sns:SetTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}
}

```

Permisos relacionados con las notificaciones en políticas administradas de solo lectura

La política administrada `AWSCodeCommitReadOnlyAccess` incluye las siguientes instrucciones para permitir el acceso de solo lectura a las notificaciones. Los usuarios con esta política administrada aplicada pueden ver notificaciones de recursos, pero no pueden crearlas, administrarlas ni suscribirse a ellas.

```
{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codecommit:*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}
```

Permisos relacionados con las notificaciones en otras políticas administradas

La política `AWSCodeCommitPowerUser` administrada incluye las siguientes instrucciones para que los usuarios puedan crear notificaciones, editarlas y suscribirse a ellas. Los usuarios no pueden eliminar reglas de notificación ni administrar etiquetas para recursos.

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
```

```

        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codecommit*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}

```

Para obtener más información acerca de IAM y las notificaciones consulte [Administración de identidades y accesos para Notificaciones de AWS CodeStar](#).

AWS CodeCommit políticas gestionadas y Amazon CodeGuru Reviewer

CodeCommit es compatible con Amazon CodeGuru Reviewer, un servicio de revisión de código automatizado que utiliza el análisis de programas y el aprendizaje automático para detectar problemas comunes y recomendar soluciones en su código Java o Python. Las políticas administradas CodeCommit incluyen declaraciones de políticas para la funcionalidad de CodeGuru Reviewer. Para obtener más información, consulte [Qué es Amazon CodeGuru Reviewer](#).

Permisos relacionados con CodeGuru Reviewer en AWSCodeCommitFullAccess

La política `AWSCodeCommitFullAccess` gestionada incluye las siguientes declaraciones para permitir que CodeGuru Reviewer se asocie y disocie de los CodeCommit repositorios. Los usuarios a los que se aplique esta política gestionada también pueden ver el estado de asociación entre CodeCommit los repositorios y CodeGuru Reviewer y ver el estado de los trabajos de revisión de las solicitudes de incorporación de datos.

```
{
  "Sid": "AmazonCodeGuruReviewerFullAccess",
  "Effect": "Allow",
  "Action": [
    "codeguru-reviewer:AssociateRepository",
    "codeguru-reviewer:DescribeRepositoryAssociation",
    "codeguru-reviewer:ListRepositoryAssociations",
    "codeguru-reviewer:DisassociateRepository",
    "codeguru-reviewer:DescribeCodeReview",
    "codeguru-reviewer:ListCodeReviews"
  ],
  "Resource": "*"
},
{
  "Sid": "AmazonCodeGuruReviewerSLRCreation",
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
  "Resource": "arn:aws:iam::*:role/aws-service-role/codeguru-reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
    }
  }
},
{
```

```

    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
        }
    }
}

```

Permisos relacionados con CodeGuru Reviewer en AWSCode CommitPowerUser

La política AWSCodeCommitPowerUser gestionada incluye las siguientes declaraciones para permitir a los usuarios asociar y desasociar repositorios con CodeGuru Reviewer, ver el estado de la asociación y ver el estado de los trabajos de revisión de las solicitudes de extracción de información.

```

{
    "Sid": "AmazonCodeGuruReviewerFullAccess",
    "Effect": "Allow",
    "Action": [
        "codeguru-reviewer:AssociateRepository",
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DisassociateRepository",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerSLRCreation",
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/codeguru-reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
    "Condition": {
        "StringLike": {

```

```

        "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
    }
}
},
{
    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
        }
    }
}
}

```

Permisos relacionados con Reviewer en CodeGuru AWSCode CommitReadOnly

La política `AWSCodeCommitReadOnlyAccess` gestionada incluye las siguientes declaraciones para permitir el acceso de solo lectura al estado de la asociación de CodeGuru revisores y ver el estado de los trabajos de revisión en el caso de las solicitudes de participación. Los usuarios a los que se ha aplicado esta política administrada no pueden asociar ni desasociar repositorios.

```

{
    "Sid": "AmazonCodeGuruReviewerReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
}

```


Función vinculada al servicio Amazon CodeGuru Reviewer

Cuando asocias un repositorio a CodeGuru Reviewer, se crea un rol vinculado a un servicio para que CodeGuru Reviewer pueda detectar problemas y recomendar soluciones para el código de Java o Python en las solicitudes de cambios. El rol vinculado a servicio se denomina `AWSServiceRoleForAmazonCodeGuruReviewer`. Para obtener más información, consulte [Uso de funciones vinculadas a servicios para Amazon CodeGuru Reviewer](#).

Para más información, consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

CodeCommit actualizaciones de las políticas gestionadas AWS

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas CodeCommit desde que este servicio comenzó a rastrear estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en [AWS CodeCommit Historial de documentos de la Guía del usuario](#).

Cambio	Descripción	Fecha
AWS política gestionada: <code>AWSCodeCommitFullAccess</code> y AWS política gestionada: <code>AWSCodeCommitPowerUser</code> : actualización de políticas existentes	CodeCommit agregó un permiso a estas políticas para admitir un tipo de notificación adicional utilizando Amazon Q Developer en aplicaciones de chat. Las <code>AWSCodeCommitFullAccess</code> políticas <code>AWSCodeCommitPowerUser</code> y se han modificado para añadir un permiso, <code>chatbot:ListMicrosoftTeamsChannelConfigurations</code>.	16 de mayo de 2023
AWS política gestionada: <code>AWSCodeCommitReadOnly</code> :	CodeCommit eliminó un permiso duplicado de la política.	18 de agosto de 2021

Cambio	Descripción	Fecha
actualización de una política actual	Se AWSCode CommitReadOnly ha modificado para eliminar un permiso duplicado <code>"iam:ListAccessKeys"</code> .	
CodeCommit comenzó a rastrear los cambios	CodeCommit comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	18 de agosto de 2021

Ejemplos de políticas administradas por el cliente

Puedes crear tus propias políticas de IAM personalizadas para permitir permisos para CodeCommit acciones y recursos. Puede asociar estas políticas personalizadas a los grupos o usuarios de IAM que requieran esos permisos. También puede crear sus propias políticas de IAM personalizadas para la integración entre CodeCommit y otros AWS servicios.

Temas

- [Ejemplos de políticas de identidad administradas por los clientes](#)

Ejemplos de políticas de identidad administradas por los clientes

El siguiente ejemplo de políticas de IAM concede permisos para diversas CodeCommit acciones. Úselas para limitar el CodeCommit acceso de sus usuarios y funciones de IAM. Estas políticas controlan la capacidad de realizar acciones con la CodeCommit consola, la API o el AWS CLI. AWS SDKs

Note

Todos los ejemplos utilizan la región EE.UU. Oeste (Oregón) (us-west-2) y contienen una cuenta ficticia. IDs

Ejemplos

- [Ejemplo 1: permitir que un usuario realice CodeCommit operaciones en una sola unidad Región de AWS](#)
- [Ejemplo 2: permitir a un usuario utilizar Git para un solo repositorio](#)
- [Ejemplo 3: permitir a un usuario conectarse desde un rango de direcciones IP específicas para obtener acceso a un repositorio](#)
- [Ejemplo 4: denegar o permitir acciones en las ramificaciones](#)
- [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#)

Ejemplo 1: permitir que un usuario realice CodeCommit operaciones en una sola unidad Región de AWS

La siguiente política de permisos utiliza un carácter comodín ("codecommit:*") para permitir a los usuarios realizar todas CodeCommit las acciones en la región us-east-2 y no desde otras. Regiones de AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codecommit:*",
      "Resource": "arn:aws:codecommit:us-east-2:111111111111:*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-2"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "codecommit:ListRepositories",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-2"
        }
      }
    }
  ]
}
```

```

    }
  ]
}
```

Ejemplo 2: permitir a un usuario utilizar Git para un solo repositorio

En CodeCommit, los permisos de la política de `GitPull` IAM se aplican a cualquier comando del cliente Git del que se recuperen datos CodeCommit `git fetch` `git clone`, incluidos, etc. Del mismo modo, los permisos de la política de `GitPush` IAM se aplican a cualquier comando del cliente Git al que se envíen los datos. CodeCommit Por ejemplo, si el permiso de política de IAM de `GitPush` está definido en `Allow`, un usuario puede enviar la eliminación de una ramificación con el protocolo Git. A ese envío no le afectan los permisos que se hayan aplicado a la operación de `DeleteBranch` para ese usuario de IAM. El `DeleteBranch` permiso se aplica a las acciones realizadas con la consola, la AWS CLI SDKs, la y la API, pero no con el protocolo Git.

El siguiente ejemplo permite al usuario especificado extraer datos del CodeCommit repositorio denominado y enviarlos a él `MyDemoRepo`:

JSON

```

{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codecommit:GitPull",
        "codecommit:GitPush"
      ],
      "Resource" : "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
    }
  ]
}
```

Ejemplo 3: permitir a un usuario conectarse desde un rango de direcciones IP específicas para obtener acceso a un repositorio

Puede crear una política que únicamente permita a los usuarios conectarse a un repositorio de CodeCommit si su dirección IP está dentro de un determinado rango de direcciones IP. Para ello,

existen dos enfoques igualmente válidos. Puede crear una Deny política que prohíba CodeCommit las operaciones si la dirección IP del usuario no está dentro de un bloque específico, o puede crear una Allow política que permita CodeCommit las operaciones si la dirección IP del usuario está dentro de un bloque específico.

Puede crear una política Deny que deniegue el acceso a todos los usuarios que no están en un determinado rango de IP. Por ejemplo, puede asociar la política administrada `AWSCodeCommitPowerUser` y una política administrada por el cliente a todos los usuarios que necesitan acceder a su repositorio. El siguiente ejemplo de política deniega todos los CodeCommit permisos a los usuarios cuyas direcciones IP no estén dentro del bloque de direcciones IP especificado de `203.0.113.0/16`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecommit:*"
      ],
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/16"
          ]
        }
      }
    }
  ]
}
```

El siguiente ejemplo de política permite al usuario especificado acceder a un CodeCommit repositorio nombrado `MyDemoRepo` con los permisos equivalentes de la política `AWSCodeCommitPowerUser` administrada solo si su dirección IP se encuentra dentro del bloque de direcciones especificado de `203.0.113.0/16`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGetRepositories",
        "codecommit:CreateBranch",
        "codecommit:CreateRepository",
        "codecommit:Get*",
        "codecommit:GitPull",
        "codecommit:GitPush",
        "codecommit:List*",
        "codecommit:Put*",
        "codecommit:Post*",
        "codecommit:Merge*",
        "codecommit:TagResource",
        "codecommit:Test*",
        "codecommit:UntagResource",
        "codecommit:Update*"
      ],
      "Resource": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/16"
          ]
        }
      }
    }
  ]
}
```

Ejemplo 4: denegar o permitir acciones en las ramificaciones

Puede crear una política que deniegue a los usuarios permisos para realizar las acciones que especifique en una o varias ramificaciones. Si lo prefiere, puede crear una política que permita realizar en una o varias ramificaciones acciones que no estén disponibles en otras ramificaciones del

repositorio. Puede utilizar estas políticas con las políticas administradas (predefinidas) apropiadas. Para obtener más información, consulte [Limite las inserciones y fusiones a las ramificaciones AWS CodeCommit](#).

Por ejemplo, puedes crear una Deny política que niegue a los usuarios la posibilidad de realizar cambios en una rama denominada main, incluida la eliminación de esa rama, en un repositorio denominado *MyDemoRepo*. Puede usar esta política con la política administrada *AWSCodeCommitPowerUser*. Los usuarios con estas dos políticas aplicadas podrían crear y eliminar ramas, crear solicitudes de extracción y todas las demás acciones según lo permita *AWSCodeCommitPowerUser*, pero no podrán insertar cambios en la rama denominada main, añadir o editar un archivo en la rama principal de la CodeCommit consola, ni fusionar ramas o una solicitud de extracción en la rama principal. Debido a que Deny se aplica a *GitPush*, debe incluir una instrucción *Null* en la política para permitir analizar la validez de las llamadas iniciales a *GitPush* cuando los usuarios lleven a cabo inserciones desde sus repositorios locales.

Tip

Si desea crear una política que se aplique a todas las ramificaciones denominadas principal de todos los repositorios de su cuenta de Amazon Web Services, para Resource, especifique un asterisco (*) en lugar del ARN de un repositorio.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecommit:GitPush",
        "codecommit>DeleteBranch",
        "codecommit:PutFile",
        "codecommit:Merge*"
      ],
      "Resource": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
      "Condition": {
        "StringEqualsIfExists": {
          "codecommit:References": [
            "refs/heads/main"
          ]
        }
      }
    }
  ]
}
```

```

        ],
      },
      "Null": {
        "codecommit:References": "false"
      }
    }
  }
]
}

```

En el siguiente ejemplo de política se permite a un usuario realizar cambios en una ramificación denominada principal en todos los repositorios de una cuenta de Amazon Web Services. No permite cambios en ninguna otra ramificación. Puedes usar esta política junto con la política `AWSCodeCommitReadOnly` administrada para permitir las transferencias automatizadas al repositorio de la rama principal. Debido a que el valor de `Effect` (Efecto) es `Allow`, esta política de ejemplo no funcionaría con las políticas administradas, como `AWSCodeCommitPowerUser`.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPush",
        "codecommit:Merge*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEqualsIfExists": {
          "codecommit:References": [
            "refs/heads/main"
          ]
        }
      }
    }
  ]
}

```


Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas

Puede crear una política que permita o deniegue acciones en los repositorios en función de las AWS etiquetas asociadas a esos repositorios y, a continuación, aplicar esas políticas a los grupos de IAM que configure para administrar los usuarios de IAM. Por ejemplo, puede crear una política que deniegue todas CodeCommit las acciones en cualquier repositorio con la clave de AWS etiqueta Status y el valor clave Secret y, a continuación, aplicar esa política al grupo de IAM que creó para desarrolladores generales (). *Developers* A continuación, debes asegurarte de que los desarrolladores que trabajan en esos repositorios etiquetados no sean miembros de ese *Developers* grupo general, sino que pertenezcan a un grupo de IAM diferente al que no se haya aplicado la política restrictiva (). SecretDevelopers

El siguiente ejemplo deniega todas CodeCommit las acciones en los repositorios etiquetados con la clave Status y el valor clave Secret:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecommit:Associate*",
        "codecommit:Batch*",
        "codecommit:CancelUploadArchive",
        "codecommit:CreateBranch",
        "codecommit:CreateCommit",
        "codecommit:CreatePullRequest*",
        "codecommit:CreateRepository",
        "codecommit:CreateUnreferencedMergeCommit",
        "codecommit>DeleteBranch",
        "codecommit>DeleteCommentContent",
        "codecommit>DeleteFile",
        "codecommit>DeletePullRequest*",
        "codecommit>DeleteRepository",
        "codecommit:Describe*",
        "codecommit:DisassociateApprovalRuleTemplateFromRepository",
        "codecommit:EvaluatePullRequestApprovalRules",
        "codecommit:GetBlob",
        "codecommit:GetBranch",
```

```

    "codecommit:GetComment*",
    "codecommit:GetCommit",
    "codecommit:GetDifferences*",
    "codecommit:GetFile",
    "codecommit:GetFolder",
    "codecommit:GetMerge*",
    "codecommit:GetObjectIdentifier",
    "codecommit:GetPullRequest*",
    "codecommit:GetReferences",
    "codecommit:GetRepository*",
    "codecommit:GetTree",
    "codecommit:GetUploadArchiveStatus",
    "codecommit:Git*",
    "codecommit:ListAssociatedApprovalRuleTemplatesForRepository",
    "codecommit:ListBranches",
    "codecommit:ListPullRequests",
    "codecommit:ListTagsForResource",
    "codecommit:Merge*",
    "codecommit:OverridePullRequestApprovalRules",
    "codecommit:Post*",
    "codecommit:Put*",
    "codecommit:TagResource",
    "codecommit:TestRepositoryTriggers",
    "codecommit:UntagResource",
    "codecommit:UpdateComment",
    "codecommit:UpdateDefaultBranch",
    "codecommit:UpdatePullRequest*",
    "codecommit:UpdateRepository*",
    "codecommit:UploadArchive"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/Status": "Secret"
    }
  }
}
]
}

```

Puede restringir aún más esta estrategia especificando repositorios concretos, en lugar de todos los repositorios, como recursos. También puedes crear políticas que permitan CodeCommit realizar

acciones en todos los repositorios que no estén etiquetados con etiquetas específicas. Por ejemplo, la siguiente política permite el equivalente a `AWSCodeCommitPowerUser` los permisos para CodeCommit las acciones, excepto que solo permite CodeCommit acciones en los repositorios que no estén etiquetados con las etiquetas especificadas:

 Note

Este ejemplo de política solo incluye acciones para CodeCommit. No incluye acciones para otros AWS servicios que están incluidos en la política `AWSCodeCommitPowerUser` gestionada. Para obtener más información, consulte [AWS política gestionada: AWSCodeCommitPowerUser](#)..

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:Associate*",
        "codecommit:Batch*",
        "codecommit:CancelUploadArchive",
        "codecommit:CreateBranch",
        "codecommit:CreateCommit",
        "codecommit:CreatePullRequest*",
        "codecommit:CreateRepository",
        "codecommit:CreateUnreferencedMergeCommit",
        "codecommit>DeleteBranch",
        "codecommit>DeleteCommentContent",
        "codecommit>DeleteFile",
        "codecommit>DeletePullRequest*",
        "codecommit:Describe*",
        "codecommit:DisassociateApprovalRuleTemplateFromRepository",
        "codecommit:EvaluatePullRequestApprovalRules",
        "codecommit:GetBlob",
        "codecommit:GetBranch",
        "codecommit:GetComment*",
        "codecommit:GetCommit",
        "codecommit:GetDifferences*",

```

```

    "codecommit:GetFile",
    "codecommit:GetFolder",
    "codecommit:GetMerge*",
    "codecommit:GetObjectIdentifier",
    "codecommit:GetPullRequest*",
    "codecommit:GetReferences",
    "codecommit:GetRepository*",
    "codecommit:GetTree",
    "codecommit:GetUploadArchiveStatus",
    "codecommit:Git*",
    "codecommit:ListAssociatedApprovalRuleTemplatesForRepository",
    "codecommit:ListBranches",
    "codecommit:ListPullRequests",
    "codecommit:ListTagsForResource",
    "codecommit:Merge*",
    "codecommit:OverridePullRequestApprovalRules",
    "codecommit:Post*",
    "codecommit:Put*",
    "codecommit:TagResource",
    "codecommit:TestRepositoryTriggers",
    "codecommit:UntagResource",
    "codecommit:UpdateComment",
    "codecommit:UpdateDefaultBranch",
    "codecommit:UpdatePullRequest*",
    "codecommit:UpdateRepository*",
    "codecommit:UploadArchive"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "aws:ResourceTag/Status": "Secret",
      "aws:ResourceTag/Team": "Saanvi"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "codecommit:CreateApprovalRuleTemplate",
    "codecommit:GetApprovalRuleTemplate",
    "codecommit:ListApprovalRuleTemplates",
    "codecommit:ListRepositories",
    "codecommit:ListRepositoriesForApprovalRuleTemplate",
    "codecommit:UpdateApprovalRuleTemplateContent",

```

```
        "codecommit:UpdateApprovalRuleTemplateDescription",
        "codecommit:UpdateApprovalRuleTemplateName"
    ],
    "Resource": "*"
}
]
```

CodeCommit referencia de permisos

En las tablas siguientes se enumeran cada operación de la CodeCommit API, las acciones correspondientes para las que puede conceder permisos y el formato del ARN del recurso que se utilizará para conceder los permisos. CodeCommit APIs Se agrupan en tablas según el alcance de las acciones permitidas por esa API. Consulte esta tabla cuando configure [Control de acceso](#) y cree políticas de permisos que vaya a asociar a una identidad de IAM (políticas basadas en identidad).

Al crear una política de permisos, debe especificar las acciones en el campo `Action` de la política. Debe especificar un valor del recurso en el campo `Resource` de la política como ARN, con o sin un carácter comodín (*).

Para expresar las condiciones en sus CodeCommit políticas, utilice claves AWS de condición generales. Para obtener una lista completa de las claves AWS de ancho ancho, consulte las [claves disponibles](#) en la Guía del usuario de IAM. Para obtener información completa sobre las acciones, los recursos y las claves de condición de CodeCommit las políticas de IAM, consulte [Acciones, recursos y claves de condición](#) de AWS CodeCommit

Note

Para especificar una acción, use el prefijo `codecommit:` seguido del nombre de la operación API (por ejemplo, `codecommit:GetRepository` o `codecommit>CreateRepository`).

Uso de comodines

Para especificar varias acciones o recursos, utilice el carácter de comodín (*) en el ARN. Por ejemplo, `codecommit:*` especifica todas CodeCommit las acciones y `codecommit:Get*` especifica todas CodeCommit las acciones que comienzan por la palabra `Get`. El siguiente ejemplo concede acceso a todos los repositorios con nombres que comienzan por `MyDemo`.

```
arn:aws:codecommit:us-west-2:111111111111:MyDemo*
```

Solo puede utilizar caracteres comodín con los *repository-name* recursos que se muestran en la tabla siguiente. No puede usar caracteres comodín con nuestros recursos *region*. *account-id* Para obtener más información acerca de los comodines, consulte [identificadores de IAM](#) en la Guía del usuario de IAM.

Temas

- [Permisos obligatorios para comandos del cliente Git](#)
- [Permisos para realizar acciones en ramificaciones](#)
- [Permisos para realizar acciones en combinaciones](#)
- [Permisos para realizar acciones en solicitudes de extracción](#)
- [Permisos para acciones en plantillas de reglas de aprobación](#)
- [Permisos para realizar acciones en archivos individuales](#)
- [Permisos para realizar acciones en comentarios](#)
- [Permisos para realizar acciones en código confirmado](#)
- [Permisos para realizar acciones en repositorios](#)
- [Permisos para realizar acciones en etiquetas](#)
- [Permisos para realizar acciones en disparadores](#)
- [Permisos para acciones de CodePipeline integración](#)

Permisos obligatorios para comandos del cliente Git

En CodeCommit, los permisos de la política de `GitPull` IAM se aplican a cualquier comando del cliente Git del que se recuperen datos CodeCommit `git fetch` `git clone`, incluidos, etc. Del mismo modo, los permisos de la política de `GitPush` IAM se aplican a cualquier comando del cliente Git al que se envíen los datos. CodeCommit Por ejemplo, si el permiso de política de IAM de `GitPush` está definido en `Allow`, un usuario puede enviar la eliminación de una ramificación con el protocolo Git. A ese envío no le afectan los permisos que se hayan aplicado a la operación de `DeleteBranch` para ese usuario de IAM. El `DeleteBranch` permiso se aplica a las acciones realizadas con la consola, la AWS CLI SDKs, la y la API, pero no con el protocolo Git.

`GitPull` y `GitPush` son permisos de políticas de IAM. No son acciones de API.

CodeCommit Permisos necesarios para las acciones de los comandos del cliente Git

GitPull

Acciones: `codecommit:GitPull`

Necesario para extraer información de un CodeCommit repositorio a un repositorio local. Se trata únicamente de un permiso de política de IAM, no de una acción de API.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GitPush

Acciones: `codecommit:Git Push`

Necesario para enviar información de un repositorio local a un CodeCommit repositorio. Se trata únicamente de un permiso de política de IAM, no de una acción de API.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en ramificaciones

Los siguientes permisos permiten o deniegan acciones en las ramas de los CodeCommit repositorios. Estos permisos se refieren únicamente a las acciones realizadas en la CodeCommit consola y con la CodeCommit API, y a los comandos ejecutados mediante la. AWS CLI No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Por ejemplo, el comando `git show-sucursal -r` muestra una lista de ramificaciones remotas para un repositorio y sus confirmaciones utilizando el protocolo de Git. No se ve afectado por ningún permiso de la CodeCommit `ListBranches` operación.

Para obtener más información sobre las políticas y ramificaciones, consulte [Limite las inserciones y fusiones a las ramificaciones AWS CodeCommit](#) y [Ejemplos de políticas administradas por el cliente](#).

CodeCommit Operaciones de la API y permisos necesarios para las acciones en las sucursales

CreateBranch

Acciones: `codecommit>CreateBranch`

Necesario para crear una rama en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[DeleteBranch](#)

Acciones: `codecommit:DeleteBranch`

Necesario para eliminar una rama de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetBranch](#)

Acciones: `codecommit:GetBranch`

Necesario para obtener detalles sobre una rama de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[ListBranches](#)

Acciones: `codecommit:ListBranches`

Necesario para obtener una lista de las sucursales de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesByFastForward](#)

Acciones: `codecommit:MergeBranchesByFastForward`

Necesario para fusionar dos sucursales mediante la estrategia de fusión rápida en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesBySquash](#)

Acciones: `codecommit:ListBranches`

Se necesita para combinar dos ramificaciones mediante la estrategia de combinación de squash en un repositorio de CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesByThreeWay](#)

Acciones: `codecommit:ListBranches`

Se necesita para combinar dos ramificaciones mediante la estrategia de fusión de tres modos en un repositorio de CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdateDefaultBranch

Acciones: `codecommit:UpdateDefaultBranch`

Necesario para cambiar la rama predeterminada en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en combinaciones

Los siguientes permisos permiten o deniegan las acciones relacionadas con las fusiones en los CodeCommit repositorios. Estos permisos se refieren a las acciones realizadas con la CodeCommit consola y la CodeCommit API, y a los comandos ejecutados mediante la. AWS CLI No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Para conocer los permisos relacionados en las ramificaciones, consulte [Permisos para realizar acciones en ramificaciones](#). Para conocer los permisos relacionados en las solicitudes de extracción, consulte [Permisos para realizar acciones en solicitudes de extracción](#).

CodeCommit Operaciones de la API y permisos necesarios para las acciones de los comandos de combinación

BatchDescribeMergeConflicts

Acciones: `codecommit:BatchDescribeMergeConflicts`

Se requiere para devolver información sobre los conflictos en una fusión entre confirmaciones de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreateUnreferencedMergeCommit

Acciones: `codecommit>CreateUnreferencedMergeCommit`

Se requiere para crear confirmaciones no referenciadas entre dos ramas o confirmaciones en un CodeCommit repositorio con el fin de compararlas e identificar posibles conflictos.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DescribeMergeConflicts

Acciones: `codecommit:DescribeMergeConflicts`

Necesario para devolver información sobre los conflictos de combinación entre las versiones base, de origen y de destino de un archivo en una posible combinación en un repositorio de CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeCommit

Acciones: `codecommit:GetMergeCommit`

Necesario para devolver información sobre la fusión entre una confirmación de origen y una de destino en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeOptions

Acciones: `codecommit:GetMergeOptions`

Se requiere para devolver información sobre las opciones de combinación disponibles entre dos ramas o especificadores de confirmación en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en solicitudes de extracción

Los siguientes permisos permiten o deniegan acciones en solicitudes de extracción de repositorios de CodeCommit. Estos permisos se refieren a las acciones realizadas con la CodeCommit consola y la CodeCommit API, y a los comandos que se ejecutan mediante la. AWS CLI No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Para conocer los permisos relacionados de comentarios, consulte [Permisos para realizar acciones en comentarios](#).

CodeCommit Operaciones de la API y permisos necesarios para realizar acciones en las solicitudes de extracción

BatchGetPullRequests

Acciones: `codecommit:BatchGetPullRequests`

Necesario para devolver información sobre una o varias solicitudes de extracción en un repositorio de CodeCommit . Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequest

Acciones: `codecommit:CreatePullRequest`

Necesario para crear una solicitud de extracción en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequestApprovalRule

Acciones: `codecommit:CreatePullRequestApprovalRule`

Se requiere para crear una regla de aprobación para una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DeletePullRequestApprovalRule

Acciones: `codecommit>DeletePullRequestApprovalRule`

Se requiere para eliminar una regla de aprobación para una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DescribePullRequestEvents

Acciones: `codecommit:DescribePullRequestEvents`

Necesario para devolver información sobre uno o más eventos de solicitudes de extracción en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

EvaluatePullRequestApprovalRules

Acciones: `codecommit:EvaluatePullRequestApprovalRules`

Necesario para evaluar si una solicitud de extracción ha cumplido todas las condiciones especificadas en sus reglas de aprobación asociadas en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentsForPullRequest

Acciones: `codecommit:GetCommentsForPullRequest`

Necesario para devolver los comentarios realizados en una solicitud de extracción.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitsFromMergeBase

Acciones: `codecommit:GetCommitsFromMergeBase`

Necesario para devolver información sobre la diferencia entre confirmaciones en el contexto de una posible combinación. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeConflicts

Acciones: `codecommit:GetMergeConflicts`

Necesario para devolver información sobre los conflictos de combinación entre la ramificación de origen y la de destino en una solicitud de extracción.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequest

Acciones: `codecommit:GetPullRequest`

Necesario para devolver información sobre una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequestApprovalStates

Acciones: `codecommit:GetPullRequestApprovalStates`

Se requiere para devolver información sobre los estados de aprobación de una solicitud de extracción especificada.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequestOverrideState

Acciones: `codecommit:GetPullRequestOverrideState`

Necesario para devolver información sobre si las reglas de aprobación se han dejado a un lado (anulado) para una solicitud de extracción y, en caso afirmativo, el nombre de recurso de Amazon (ARN) del usuario o la identidad que anuló las reglas, así como sus requisitos para la solicitud de extracción.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

ListPullRequests

Acciones: `codecommit:ListPullRequests`

Necesario para ver las solicitudes de extracción en un repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

MergePullRequestByFastForward

Acciones: `codecommit:MergePullRequestByFastForward`

Necesario para cerrar una solicitud de extracción e intentar combinar la ramificación de origen en la ramificación de destino de una solicitud de extracción utilizando la estrategia de combinación de avance rápido.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

MergePullRequestBySquash

Acciones: `codecommit:MergePullRequestBySquash`

Necesario para cerrar una solicitud de extracción e intentar combinar la ramificación de origen en la ramificación de destino de una solicitud de extracción utilizando la estrategia de combinación de squash.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergePullRequestByThreeWay](#)

Acciones: `codecommit:MergePullRequestByThreeWay`

Necesario para cerrar una solicitud de extracción e intentar combinar la ramificación de origen en la ramificación de destino de una solicitud de extracción utilizando la estrategia de combinación de tres modos.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[OverridePullRequestApprovalRules](#)

Acciones: `codecommit:OverridePullRequestApprovalRules`

Se requiere dejar de lado todos los requisitos de las reglas de aprobación para una solicitud de extracción en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[PostCommentForPullRequest](#)

Acciones: `codecommit:PostCommentForPullRequest`

Necesario para publicar un comentario en una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestApprovalRuleContent](#)

Acciones: `codecommit:UpdatePullRequestApprovalRuleContent`

Necesario para cambiar la estructura de una regla de aprobación de una solicitud de extracción en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestApprovalState](#)

Acciones: `codecommit:UpdatePullRequestApprovalState`

Se requiere para actualizar el estado de una aprobación en una solicitud de extracción en un repositorio de CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestDescription](#)

Acciones: `codecommit:UpdatePullRequestDescription`

Necesario para cambiar la descripción de una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestStatus](#)

Acciones: `codecommit:UpdatePullRequestStatus`

Necesario para cambiar el estado de una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestTitle](#)

Acciones: `codecommit:UpdatePullRequestTitle`

Necesario para cambiar el título de una solicitud de extracción en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para acciones en plantillas de reglas de aprobación

Los siguientes permisos permiten o deniegan acciones en plantillas de reglas de aprobación en repositorios de CodeCommit. Estos permisos se refieren únicamente a las acciones realizadas en la CodeCommit consola, la CodeCommit API y a los comandos ejecutados mediante la AWS CLI. No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Para conocer los permisos relacionados en las solicitudes de extracción, consulte [Permisos para realizar acciones en solicitudes de extracción](#).

CodeCommit Operaciones de API y permisos necesarios para realizar acciones en las plantillas de reglas de aprobación

[AssociateApprovalRuleTemplateWithRepository](#)

Acciones: `codecommit:AssociateApprovalRuleTemplateWithRepository`

Necesario para asociar una plantilla con un repositorio especificado en una cuenta de Amazon Web Services. Una vez asociado, se crean automáticamente reglas de aprobación que coinciden con las condiciones de plantilla en cada solicitud de extracción creada en el repositorio especificado.

Recurso: *

[BatchAssociateApprovalRuleTemplateWithRepositories](#)

Acciones: `codecommit:BatchAssociateApprovalRuleTemplateWithRepositories`

Necesario para asociar una plantilla con uno o varios repositorios especificados en una cuenta de Amazon Web Services.

Recurso: *

[BatchDisassociateApprovalRuleTemplateFromRepositories](#)

Acciones: `codecommit:BatchDisassociateApprovalRuleTemplateFromRepositories`

Necesario para desasociar una plantilla de uno o varios repositorios especificados en una cuenta de Amazon Web Services.

Recurso: *

[CreateApprovalRuleTemplate](#)

Acciones: `codecommit>CreateApprovalRuleTemplate`

Necesario para crear una plantilla de regla de aprobación que pueda asociarse a uno o más repositorios de la cuenta de Amazon Web Services.

Recurso: *

[DeleteApprovalRuleTemplate](#)

Acciones: `codecommit>DeleteApprovalRuleTemplate`

Se requiere para eliminar una plantilla de regla de aprobación de una cuenta de AWS .

Recurso: *

[DisassociateApprovalRuleTemplateFromRepository](#)

Acciones: `codecommit:DisassociateApprovalRuleTemplateFromRepository`

Necesario para desasociar la plantilla especificada de un repositorio en una cuenta de Amazon Web Services. No elimina las reglas de aprobación en las solicitudes de extracción ya creadas con la plantilla.

Recurso: *

[GetApprovalRuleTemplate](#)

Acciones: `codecommit:GetApprovalRuleTemplate`

Necesario para devolver información sobre una plantilla de regla de aprobación en una cuenta de Amazon Web Services.

Recurso: *

[ListApprovalRuleTemplates](#)

Acciones: `codecommit:ListApprovalRuleTemplates`

Necesario para mostrar plantillas de reglas de aprobación en una cuenta de Amazon Web Services.

Recurso: *

[ListAssociatedApprovalRuleTemplatesForRepository](#)

Acciones: `codecommit:ListAssociatedApprovalRuleTemplatesForRepository`

Necesario para enumerar todas las plantillas de reglas de aprobación asociadas a un repositorio especificado en una cuenta de Amazon Web Services.

Recurso: *

[ListRepositoriesForApprovalRuleTemplate](#)

Acciones: `codecommit:ListRepositoriesForApprovalRuleTemplate`

Necesario para enumerar todos los repositorios asociados a una plantilla de regla de aprobación especificada en una cuenta de Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateContent](#)

Acciones: `codecommit:UpdateApprovalRuleTemplateContent`

Necesario para actualizar el contenido de una plantilla de regla de aprobación en una cuenta de Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateDescription](#)

Acciones: `codecommit:UpdateApprovalRuleTemplateDescription`

Necesario para actualizar la descripción de una plantilla de regla de aprobación en una cuenta de Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateName](#)

Acciones: `codecommit:UpdateApprovalRuleTemplateName`

Se requiere para actualizar el nombre de una plantilla de regla de aprobación en una cuenta de AWS .

Recurso: *

Permisos para realizar acciones en archivos individuales

Los siguientes permisos permiten o deniegan acciones en archivos individuales de repositorios de CodeCommit. Estos permisos se refieren únicamente a las acciones realizadas en la CodeCommit consola, la CodeCommit API y a los comandos ejecutados mediante la AWS CLI. No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Por ejemplo, el comando `git push` envía archivos nuevos y modificados a un repositorio de CodeCommit mediante el protocolo de Git. No se ve afectado por ningún permiso de la CodeCommit `PutFile` operación.

CodeCommit Operaciones de la API y permisos necesarios para realizar acciones en archivos individuales

[DeleteFile](#)

Acciones: `codecommit>DeleteFile`

Necesario para eliminar un archivo específico de una rama específica de un CodeCommit repositorio desde la CodeCommit consola.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetBlob

Acciones: `codecommit:GetBlob`

Necesario para ver el contenido codificado de un archivo individual en un CodeCommit repositorio desde la CodeCommit consola.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetFile

Acciones: `codecommit:GetFile`

Necesario para ver el contenido codificado de un archivo específico y sus metadatos en un CodeCommit repositorio desde la CodeCommit consola.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetFolder

Acciones: `codecommit:GetFolder`

Necesario para ver el contenido de una carpeta específica en un CodeCommit repositorio desde la CodeCommit consola.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PutFile

Acciones: `codecommit:PutFile`

Necesario para añadir un archivo nuevo o modificado a un CodeCommit repositorio desde la CodeCommit consola, la CodeCommit API o el AWS CLI.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en comentarios

Los siguientes permisos permiten o deniegan acciones en los comentarios de CodeCommit los repositorios. Estos permisos se refieren a las acciones realizadas con la CodeCommit consola y la CodeCommit API, y a los comandos ejecutados mediante la. AWS CLI Para conocer los permisos relacionados de comentarios en solicitudes de extracción, consulte [Permisos para realizar acciones en solicitudes de extracción](#).

CodeCommit Operaciones de la API y permisos necesarios para realizar acciones en los repositorios

[DeleteCommentContent](#)

Acciones: `codecommit:DeleteCommentContent`

Necesario para eliminar el contenido de un comentario realizado en un cambio, archivo o confirmación en un repositorio. Los comentarios no se pueden eliminar, pero si el usuario tiene este permiso, puede borrar el contenido de un comentario.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetComment](#)

Acciones: `codecommit:GetComment`

Se requiere para devolver información sobre un comentario realizado en un cambio, un archivo o una confirmación en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommentReactions](#)

Acciones: `codecommit:GetCommentReactions`

Se requiere para devolver información sobre las reacciones de los emojis a un comentario realizado en un cambio, un archivo o una confirmación en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommentsForComparedCommit](#)

Acciones: `codecommit:GetCommentsForComparedCommit`

Se requiere para devolver información sobre los comentarios realizados en la comparación entre dos confirmaciones de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[PostCommentForComparedCommit](#)

Acciones: `codecommit:PostCommentForComparedCommit`

Necesario para realizar un comentario en la comparación entre dos confirmaciones en un repositorio de CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[PostCommentReply](#)

Acciones: `codecommit:PostCommentReply`

Se requiere para crear una respuesta a un comentario sobre una comparación entre confirmaciones o sobre una solicitud de cambios en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[PutCommentReaction](#)

Acciones: `codecommit:PutCommentReaction`

Se requiere para responder a un comentario con un emoji en una confirmación o en una solicitud de extracción en un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdateComment](#)

Acciones: `codecommit:UpdateComment`

Necesario para editar un comentario en una comparación entre confirmaciones o en una solicitud de extracción. Los comentarios solo puede editarlos el autor del comentario.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en código confirmado

Los siguientes permisos permiten o deniegan acciones en el código confirmado en repositorios de CodeCommit. Estos permisos se refieren a las acciones realizadas con la CodeCommit consola y la CodeCommit API, y a los comandos ejecutados mediante la AWS CLI. No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git. Por ejemplo, el comando `git commit` crea una confirmación de una ramificación en un repositorio utilizando el protocolo de Git. No se ve afectado por ningún permiso de la CodeCommit `CreateCommit` operación.

La denegación explícita de algunos de estos permisos puede tener consecuencias inesperadas en la CodeCommit consola. Por ejemplo, si define `GetTree` en Deny, evitará que los usuarios examinen el contenido de un repositorio en la consola, pero no impedirá que los usuarios vean el contenido de un archivo en el repositorio (por ejemplo, si reciben un enlace al archivo por correo

electrónico). Si define `GetBlob` en `Deny`, evitará que los usuarios vean el contenido de los archivos, pero no impedirá que los usuarios examinen la estructura de un repositorio. Si define `GetCommit` en `Deny`, evitará que los usuarios recuperen información detallada sobre las confirmaciones. Si define `GetObjectIdentifier` en `Deny`, se bloqueará la mayor parte de la funcionalidad de exploración de código. Si estableces estas tres acciones `Deny` en una política, un usuario con esa política no podrá navegar por el código de la CodeCommit consola.

CodeCommit Operaciones de la API y permisos necesarios para realizar acciones en el código comprometido

BatchGetCommits

Acciones: `codecommit:BatchGetCommits`

Necesario para devolver información sobre una o varias confirmaciones en un repositorio de CodeCommit. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[CreateCommit](#)

Acciones: `codecommit>CreateCommit`

Se necesita para crear una confirmación.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommit](#)

Acciones: `codecommit:GetCommit`

Necesario para devolver información sobre una confirmación.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitHistory

Acciones: `codecommit:GetCommitHistory`

Necesario para devolver información sobre el historial de confirmaciones en un repositorio. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetDifferences

Acciones: `codecommit:GetDifferences`

Necesario para devolver información sobre las diferencias en un especificador de confirmación (como, por ejemplo, una ramificación, etiqueta, HEAD, ID de confirmación u otra referencia completa).

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetObjectIdentifier

Acciones: `codecommit:GetObjectIdentifier`

Necesario para resolver blobs, árboles y confirmaciones con su identificador. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetReferences

Acciones: `codecommit:GetReferences`

Necesario para devolver todas las referencias, como ramificaciones y etiquetas. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetTree

Acciones: `codecommit:GetTree`

Necesario para ver el contenido de un árbol específico en un CodeCommit repositorio desde la CodeCommit consola. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en repositorios

Los siguientes permisos permiten o deniegan acciones en CodeCommit los repositorios. Estos permisos se refieren a las acciones realizadas con la CodeCommit consola y la CodeCommit API, y a los comandos ejecutados mediante la. AWS CLI No tienen que ver con acciones similares que se pueden llevar a cabo utilizando el protocolo de Git.

CodeCommit Operaciones de la API y permisos necesarios para realizar acciones en los repositorios

[BatchGetRepositories](#)

Acciones: `codecommit:BatchGetRepositories`

Necesario para obtener información sobre varios CodeCommit repositorios que se encuentran en una cuenta de Amazon Web Services. En `Resource`, debe especificar los nombres de todos los CodeCommit repositorios para los que se permite (o deniega) información a un usuario.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[CreateRepository](#)

Acciones: `codecommit:CreateRepository`

Necesario para crear un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[DeleteRepository](#)

Acciones: `codecommit>DeleteRepository`

Necesario para eliminar un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetRepository](#)

Acciones: `codecommit:GetRepository`

Necesario para obtener información sobre un único CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[ListRepositories](#)

Acciones: `codecommit:ListRepositories`

Necesario para obtener una lista de los nombres y el sistema IDs de varios CodeCommit repositorios de una cuenta de Amazon Web Services. El único valor permitido para `Resource` para esta acción es todos los repositorios (*).

Recurso: *

[UpdateRepositoryDescription](#)

Acciones: `codecommit:UpdateRepositoryDescription`

Necesario para cambiar la descripción de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdateRepositoryName](#)

Acciones: `codecommit:UpdateRepositoryName`

Necesario para cambiar el nombre de un CodeCommit repositorio. En `Resource`, debe especificar tanto los CodeCommit repositorios que se pueden cambiar como los nombres de los nuevos repositorios.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en etiquetas

Los siguientes permisos permiten o deniegan acciones en las AWS etiquetas de los CodeCommit recursos.

CodeCommit Operaciones de la API y permisos necesarios para las acciones en las etiquetas

[ListTagsForResource](#)

Acciones: `codecommit:ListTagsForResource`

Se requiere para devolver información sobre las AWS etiquetas configuradas en un recurso en CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[TagResource](#)

Acciones: `codecommit:TagResource`

Necesario para añadir o editar AWS etiquetas para un repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UntagResource](#)

Acciones: `codecommit:UntagResource`

Necesario para eliminar AWS etiquetas de un recurso en CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para realizar acciones en disparadores

Los siguientes permisos permiten o deniegan acciones en disparadores de repositorios de CodeCommit.

CodeCommit Operaciones de la API y permisos necesarios para las acciones en los desencadenadores

[GetRepositoryTriggers](#)

Acciones: `codecommit:GetRepositoryTriggers`

Necesario para devolver información sobre los disparadores configurados para un repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[PutRepositoryTriggers](#)

Acciones: `codecommit:PutRepositoryTriggers`

Necesario para crear, editar o eliminar disparadores de un repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[TestRepositoryTriggers](#)

Acciones: `codecommit:TestRepositoryTriggers`

Necesario para probar las funciones de un disparador de repositorio mediante el envío de datos al tema o a la función configurados para el disparador.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permisos para acciones de CodePipeline integración

CodePipeline Para utilizar un CodeCommit repositorio en una acción de origen para una canalización, debes conceder todos los permisos que se indican en la siguiente tabla al rol de servicio CodePipeline. Si no se han establecido estos permisos en la función de servicio o se han

establecido en **Deny**, la canalización no se ejecuta automáticamente cuando se realiza un cambio en el repositorio, y los cambios no se podrán liberar manualmente.

CodeCommit Operaciones de la API y permisos necesarios para las acciones de integración
CodePipeline

GetBranch

Acciones: `codecommit:GetBranch`

Necesario para obtener detalles sobre una rama de un CodeCommit repositorio.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommit

Acciones: `codecommit:GetCommit`

Necesario para devolver información sobre una confirmación.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UploadArchive

Acciones: `codecommit:UploadArchive`

Necesario para permitir que la función de servicio cargue los cambios del repositorio en una canalización. CodePipeline Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetUploadArchiveStatus

Acciones: `codecommit:GetUploadArchiveStatus`

Necesario para determinar el estado de la carga de un archivo: si está en curso, si se ha completado, si se ha cancelado o si se ha producido un error. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CancelUploadArchive

Acciones: `codecommit:CancelUploadArchive`

Necesario para cancelar la carga de un archivo en una canalización. Se trata únicamente de un permiso de política de IAM, no de una acción de API a la que se puede llamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

¿Cómo AWS CodeCommit funciona con IAM

Antes de utilizar IAM para gestionar el acceso CodeCommit, debe comprender las funciones de IAM disponibles para su uso. CodeCommit Para obtener una visión general de cómo funcionan con IAM CodeCommit y otros AWS servicios, consulte [AWS Servicios que funcionan con IAM en la Guía del usuario de IAM](#).

Temas

- [Claves de condición](#)
- [Ejemplos](#)

Claves de condición

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como `igual` o `menor que`, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

CodeCommit define su propio conjunto de claves de condición y también admite el uso de algunas claves de condición globales. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

Algunas CodeCommit acciones admiten la clave de `codecommit:References` condición. Para obtener un ejemplo de política que utiliza esta clave, consulte [Ejemplo 4: denegar o permitir acciones en las ramificaciones](#).

Para ver una lista de claves de CodeCommit condición, consulte las [claves de condición AWS CodeCommit](#) en la Guía del usuario de IAM. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte [Acciones definidas por AWS CodeCommit](#).

Ejemplos

Para ver ejemplos de políticas CodeCommit basadas en la identidad, consulte [AWS CodeCommit ejemplos de políticas basadas en la identidad](#)

CodeCommit políticas basadas en recursos

CodeCommit no admite políticas basadas en recursos.

Autorización basada en etiquetas CodeCommit

Puede adjuntar etiquetas a CodeCommit los recursos o pasarles etiquetas en una solicitud CodeCommit. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `codecommit:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Para obtener más información sobre el etiquetado de CodeCommit recursos, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#). Para obtener más información sobre las estrategias de etiquetado, consulte Recursos de [etiquetado AWS](#).

CodeCommit también admite políticas basadas en etiquetas de sesión. Para obtener más información, consulte [Etiquetas de sesión](#).

Uso de etiquetas para proporcionar información de identidad en CodeCommit

CodeCommit admite el uso de etiquetas de sesión, que son atributos de pares clave-valor que se transfieren cuando se asume una función de IAM, se utilizan credenciales temporales o se federa un usuario en (). AWS Security Token Service AWS STS También puede asociar etiquetas a un usuario de IAM. Puede utilizar la información proporcionada en estas etiquetas para facilitar la identificación de quién ha realizado un cambio o ha provocado un evento. CodeCommit incluye los valores de las etiquetas con los siguientes nombres clave en CodeCommit los eventos:

Nombre de la clave	Valor
<code>displayName</code>	Nombre legible por humanos para mostrar y asociar con el usuario (por ejemplo, Mary Major o Saanvi Sarkar).
<code>emailAddress</code>	La dirección de correo electrónico que desea mostrar y asociar al usuario (por ejemplo,

Nombre de la clave	Valor
	mary_major@example.com o saanvi_sarkar@example.com).

Si se proporciona esta información, CodeCommit inclúyala en los eventos enviados a Amazon EventBridge y Amazon CloudWatch Events. Para obtener más información, consulte [Supervisión de eventos de CodeCommit en Amazon EventBridge y eventos de Amazon CloudWatch](#).

Para utilizar el etiquetado de sesión, los roles deben tener políticas que incluyan el permiso `sts:TagSession` establecido en `Allow`. Si está utilizando acceso federado, puede configurar la información de nombre de visualización y etiqueta de correo electrónico como parte de la configuración. Por ejemplo, si utiliza Azure Active Directory, puede proporcionar la siguiente información de reclamación:

Nombre de la reclamación	Valor
<code>https://aws.amazon.com/SAML/Attributes/PrincipalTag:displayName</code>	<code>user.displayName</code>
<code>https://aws.amazon.com/SAML/Attributes/PrincipalTag:emailAddress</code>	<code>user.mail</code>

Puede utilizar las etiquetas de sesión AWS CLI para pasarse `displayName` y `emailAddress` utilizarlas `AssumeRole`. Por ejemplo, un usuario que quiera asumir un rol llamado y *Developer* que quiera asociar su nombre *Mary Major* podría usar un `assume-role` comando similar al siguiente:

```
aws sts assume-role \  
--role-arn arn:aws:iam::123456789012:role/Developer \  
--role-session-name Mary-Major \  
--tags Key=displayName,Value="Mary Major" \  
Key=emailAddress,Value="mary_major@example.com" \  
--external-id Example987
```

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

Puede utilizar la operación `AssumeRoleWithSAML` para devolver un conjunto de credenciales temporales que incluyan etiquetas `displayName` y `emailAddress`. Puede utilizar estas etiquetas cuando acceda a repositorios de CodeCommit. Para ello, es necesario que su empresa o grupo ya haya integrado la solución SAML de terceros con AWS ella. Si es así, puede pasar atributos SAML como etiquetas de sesión. Por ejemplo, si desea transferir los atributos de identidad de un nombre visible y una dirección de correo electrónico a un usuario *Saanvi Sarkar* denominado etiquetas de sesión:

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:displayName">
  <AttributeValue>Saarvi Sarkar</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:emailAddress">
  <AttributeValue>saanvi_sarkar@example.com</AttributeValue>
</Attribute>
```

Para obtener más información, consulte [Pasar etiquetas de sesión mediante AssumeRoleWithSAML](#).

Puede utilizar la operación `AssumeRoleWithIdentity` para devolver un conjunto de credenciales temporales que incluyan etiquetas `displayName` y `emailAddress`. Puede utilizar estas etiquetas cuando acceda a repositorios de CodeCommit. Para pasar las etiquetas de sesión desde OpenID Connect (OIDC), debe incluir las etiquetas de sesión en JSON Web Token (JWT). Por ejemplo, el token JWP decodificado `AssumeRoleWithWebIdentity` que se utiliza para realizar la llamada incluye las etiquetas `displayName` y de `emailAddress` sesión de un usuario llamado: *Li Juan*

```
{
  "sub": "lijuan",
  "aud": "ac_oic_client",
  "jti": "ZYUCeREXAMPLE",
  "iss": "https://xyz.com",
  "iat": 1566583294,
  "exp": 1566583354,
  "auth_time": 1566583292,
  "https://aws.amazon.com/tags": {
    "principal_tags": {
      "displayName": ["Li Juan"],
      "emailAddress": ["li_juan@example.com"],
    },
    "transitive_tag_keys": [
      "displayName",
      "emailAddress"
    ]
  }
}
```

```
]
}
}
```

Para obtener más información, consulte [Pasar etiquetas de sesión mediante](#).

`AssumeRoleWithWebIdentity`

Puede utilizar la operación `GetFederationToken` para devolver un conjunto de credenciales temporales que incluyan etiquetas `displayName` y `emailAddress`. Puede utilizar estas etiquetas cuando acceda a repositorios de CodeCommit. Por ejemplo, para usar el y AWS CLI obtener un token de federación que incluya las `emailAddress` etiquetas `displayName` y:

```
aws sts get-federation-token \
--name my-federated-user \
--tags key=displayName,value="Nikhil Jayashankar"
      key=emailAddress,value=nikhil_jayashankar@example.com
```

Para obtener más información, consulte [Pasar etiquetas de sesión mediante `GetFederationToken`](#).

CodeCommit Funciones de IAM

Un [rol de IAM](#) es una entidad de la cuenta de Amazon Web Services que dispone de permisos específicos.

Uso de credenciales temporales con CodeCommit

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen llamando a operaciones de AWS STS API como [AssumeRole](#) o [GetFederationToken](#).

CodeCommit admite el uso de credenciales temporales. Para obtener más información, consulte [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#).

Roles vinculados a servicios

Los [roles vinculados a un servicio](#) permiten a AWS los servicios acceder a los recursos de otros servicios para completar una acción en tu nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

CodeCommit no utiliza funciones vinculadas al servicio.

Roles de servicio

Esta característica permite que un servicio asuma un [rol de servicio](#) en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

CodeCommit no utiliza funciones de servicio.

AWS CodeCommit ejemplos de políticas basadas en la identidad

De forma predeterminada, los usuarios y los roles de IAM no tienen permiso para crear, ver ni modificar recursos de CodeCommit. Tampoco pueden realizar tareas con la API Consola de administración de AWS AWS CLI, o AWS. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe asociar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para ver políticas de ejemplo, consulte lo siguiente:

- [Ejemplo 1: permitir que un usuario realice CodeCommit operaciones en una sola unidad Región de AWS](#)
- [Ejemplo 2: permitir a un usuario utilizar Git para un solo repositorio](#)
- [Ejemplo 3: permitir a un usuario conectarse desde un rango de direcciones IP específicas para obtener acceso a un repositorio](#)
- [Ejemplo 4: denegar o permitir acciones en las ramificaciones](#)
- [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#)
- [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#)

Para obtener información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas en la pestaña JSON](#) en la Guía del usuario de IAM.

Temas

- [Prácticas recomendadas relativas a políticas](#)
- [Uso de la consola CodeCommit](#)

- [Permitir a los usuarios consultar sus propios permisos](#)
- [Visualización CodeCommit repositories basada en etiquetas](#)

Prácticas recomendadas relativas a políticas

Las políticas basadas en la identidad determinan si alguien puede crear CodeCommit recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de tarea](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte [Validación de políticas con el Analizador de acceso de IAM](#) en la Guía del usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte [Acceso seguro a la API con MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Uso de la consola CodeCommit

Para acceder a la AWS CodeCommit consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los CodeCommit recursos de su cuenta de Amazon Web Services. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles de IAM) que tengan esa política.

Para garantizar que esas entidades puedan seguir utilizando la CodeCommit consola, adjunte también la siguiente política AWS gestionada a las entidades. Para obtener más información, consulte [Agregar de permisos a un usuario](#) en la Guía del usuario de IAM.

Para obtener más información, consulte [Uso de políticas basadas en la identidad \(políticas de IAM\) para CodeCommit](#).

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
```

```

        "Action": [
            "iam:GetUserPolicy",
            "iam:ListGroupsForUser",
            "iam:ListAttachedUserPolicies",
            "iam:ListUserPolicies",
            "iam:GetUser"
        ],
        "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
        "Sid": "NavigateInConsole",
        "Effect": "Allow",
        "Action": [
            "iam:GetGroupPolicy",
            "iam:GetPolicyVersion",
            "iam:GetPolicy",
            "iam:ListAttachedGroupPolicies",
            "iam:ListGroupPolicies",
            "iam:ListPolicyVersions",
            "iam:ListPolicies",
            "iam:ListUsers"
        ],
        "Resource": "*"
    }
]
}

```

Visualización CodeCommit *repositories* basada en etiquetas

Puede utilizar las condiciones de su política basada en identidad para controlar el acceso a los recursos de CodeCommit basados en etiquetas. Para obtener un ejemplo de política que muestra cómo hacerlo, consulte [Ejemplo 5: denegar o permitir acciones en repositorios con etiquetas](#).

Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.

Solución de problemas AWS CodeCommit de identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con un CodeCommit IAM.

Temas

- [No estoy autorizado a realizar ninguna acción en CodeCommit](#)
- [No estoy autorizado a realizar lo siguiente: PassRole](#)
- [Quiero ver mis claves de acceso](#)
- [Soy administrador y quiero permitir que otras personas accedan CodeCommit](#)
- [Quiero permitir que personas ajenas a mi cuenta de Amazon Web Services accedan a mis CodeCommit recursos](#)

No estoy autorizado a realizar ninguna acción en CodeCommit

Si Consola de administración de AWS le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Para obtener más información, consulte [Permisos necesarios para usar la consola CodeCommit](#)

No estoy autorizado a realizar lo siguiente: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a CodeCommit.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada al servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en CodeCommit. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero ver mis claves de acceso

Después de crear sus claves de acceso de usuario de IAM, puede ver su ID de clave de acceso en cualquier momento. Sin embargo, no puede volver a ver su clave de acceso secreta. Si pierde la clave de acceso secreta, debe crear un nuevo par de claves de acceso.

Las claves de acceso se componen de dos partes: un ID de clave de acceso (por ejemplo, AKIAIOSFODNN7EXAMPLE) y una clave de acceso secreta (por ejemplo, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). El ID de clave de acceso y la clave de acceso secreta se utilizan juntos, como un nombre de usuario y contraseña, para autenticar sus solicitudes. Administre sus claves de acceso con el mismo nivel de seguridad que para el nombre de usuario y la contraseña.

Important

No proporcione las claves de acceso a terceros, ni siquiera para que lo ayuden a [buscar el ID de usuario canónico](#). De este modo, podrías dar a alguien acceso permanente a tu Cuenta de AWS.

Cuando crea un par de claves de acceso, se le pide que guarde el ID de clave de acceso y la clave de acceso secreta en un lugar seguro. La clave de acceso secreta solo está disponible en el momento de su creación. Si pierde la clave de acceso secreta, debe agregar nuevas claves de acceso a su usuario de IAM. Puede tener un máximo de dos claves de acceso. Si ya cuenta con dos, debe eliminar un par de claves antes de crear una nueva. Para consultar las instrucciones, consulte [Administración de claves de acceso](#) en la Guía del usuario de IAM.

Soy administrador y quiero permitir que otras personas accedan CodeCommit

Para permitir el acceso de otras personas CodeCommit, debes conceder permiso a las personas o aplicaciones que necesitan acceso. Si usa AWS IAM Identity Center para administrar las personas y las aplicaciones, debe asignar conjuntos de permisos a los usuarios o grupos para definir su nivel de acceso. Los conjuntos de permisos crean políticas de IAM y las asignan a los roles de IAM asociados a la persona o aplicación de forma automática. Para obtener más información, consulte la sección [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

Si no utiliza IAM Identity Center, debe crear entidades de IAM (usuarios o roles) para las personas o aplicaciones que necesitan acceso. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en CodeCommit. Una vez concedidos los permisos, proporcione las

credenciales al usuario o al desarrollador de la aplicación. Utilizarán esas credenciales para acceder a AWS. Para obtener más información sobre la creación de usuarios, grupos, políticas y permisos de IAM, consulte [Identidades de IAM](#) y [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.

Quiero permitir que personas ajenas a mi cuenta de Amazon Web Services accedan a mis CodeCommit recursos

Para obtener más información, consulte [Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles](#).

Resiliencia en AWS CodeCommit

La infraestructura global de AWS se divide en Regiones de AWS y zonas de disponibilidad. Las Regiones de AWS 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 latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que conmutan automáticamente entre zonas sin interrupción. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Existe un repositorio de CodeCommit o una plantilla de reglas de aprobación de CodeCommit en la Región de AWS donde se creó. Para obtener más información, consulte [Regiones y puntos de conexión de Git para AWS CodeCommit](#). Para mejorar la resiliencia de los repositorios, puede configurar su cliente Git para que lo suba a dos repositorios a la vez. Para obtener más información, consulte [Envío de confirmaciones a un repositorio de Git adicional](#).

Para obtener más información sobre las Regiones de AWS y las zonas de disponibilidad, consulte [Infraestructura global de AWS](#).

Seguridad de la infraestructura en () AWS CodeCommit

Al tratarse de un servicio administrado, AWS CodeCommit está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el documento técnico [Amazon Web Services: Información general sobre los procesos de seguridad](#).

Puede utilizar llamadas a la API publicadas en AWS para acceder a CodeCommit a través de la red. Los clientes deben ser compatibles con la seguridad de la capa de transporte (TLS) 1.0 o una versión posterior. Recomendamos TLS 1.2 o una versión posterior. Los clientes también

deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

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.

Puede llamar a estas operaciones de API desde cualquier ubicación de red, pero CodeCommit admite restricciones basadas en la dirección IP de origen. También puede utilizar políticas de CodeCommit para controlar el acceso desde puntos de conexión específicos de Amazon Virtual Private Cloud (Amazon VPC) o VPC específicas. Este proceso aísla con eficacia el acceso de red a un recurso de CodeCommit determinado únicamente desde la VPC específica de la red de AWS.

Para obtener más información, consulte los siguientes temas:

- [Ejemplo 1: permitir que un usuario realice CodeCommit operaciones en una sola unidad Región de AWS](#)
- [Ejemplo 3: permitir a un usuario conectarse desde un rango de direcciones IP específicas para obtener acceso a un repositorio](#)
- [Utilización AWS CodeCommit con puntos finales de VPC de interfaz](#)

Supervisión de AWS CodeCommit

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS y de sus otras soluciones de CodeCommit. AWS ofrece las siguientes herramientas de supervisión para vigilar CodeCommit, informar cuando algo no va bien y tomar medidas automáticamente cuando proceda:

- Amazon EventBridge puede utilizarse para automatizar los servicios de AWS y responder automáticamente a eventos del sistema, como problemas de disponibilidad de las aplicaciones o cambios en los recursos. Los eventos de los servicios de AWS se envían a EventBridge casi en tiempo real. Puede crear reglas sencillas para indicar qué eventos le resultan de interés, así como qué acciones automatizadas se van a realizar cuando un evento cumple una de las reglas. Para obtener más información, consulte la [Guía del usuario de Amazon EventBridge](#) y [Supervisión de eventos de CodeCommit en Amazon EventBridge y eventos de Amazon CloudWatch](#).
- Los eventos de Amazon CloudWatch proporcionan un flujo de eventos de sistema casi en tiempo real que describe cambios en los recursos de AWS. CloudWatch Events habilita la computación basada en eventos automatizada, para que pueda escribir reglas que vigilan determinados eventos y desencadenan acciones automatizadas en otros servicios de AWS cuando estos eventos se producen. Para obtener más información, consulte la [Guía del usuario de los eventos de Amazon CloudWatch](#) y [Supervisión de eventos de CodeCommit en Amazon EventBridge y eventos de Amazon CloudWatch](#).
- Registros de Amazon CloudWatch: se puede usar para supervisar, almacenar y acceder a sus archivos de registro de CloudTrail u otras fuentes. Los registros de CloudWatch pueden supervisar la información de los archivos de registro y enviarle una notificación cuando se alcancen determinados umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulte la [Guía del usuario de Registros de Amazon CloudWatch](#).
- AWS CloudTrail captura llamadas a la API y eventos relacionados efectuados por su cuenta de Amazon Web Services o en su nombre, y entrega los archivos de registro al bucket de Amazon S3 que se haya especificado. También puede identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para obtener más información, consulte la [Guía de usuario de AWS CloudTrail](#) y [Registrar llamadas a la AWS CodeCommit API con AWS CloudTrail](#).

Supervisión de eventos de CodeCommit en Amazon EventBridge y eventos de Amazon CloudWatch

Puede supervisar los eventos de la AWS CodeCommit en EventBridge, lo que envía un flujo de datos en tiempo real desde sus propias aplicaciones, aplicaciones de software como servicio (SaaS) y servicios de AWS. EventBridge dirige esos datos a los objetivos, como AWS Lambda y Amazon Simple Notification Service. Estos eventos son los mismos que los que aparecen en Eventos de Amazon CloudWatch, lo que proporciona un flujo de eventos del sistema casi en tiempo real que describen los cambios en los recursos de AWS.

Los siguientes ejemplos corresponden a eventos para CodeCommit.

Note

CodeCommit permite facilitar la información de `displayName` y `emailAddress` incluida en las etiquetas de sesión en eventos, si esa información está disponible. Para obtener más información, consulte [Etiquetas de sesión](#) y [Uso de etiquetas para proporcionar información de identidad en CodeCommit](#).

Temas

- [Evento `referenceCreated`](#)
- [Evento `referenceUpdated`](#)
- [Evento `referenceDeleted`](#)
- [Evento `unreferencedMergeCommitCreated`](#)
- [Evento `commentOnCommitCreated`](#)
- [Evento `commentOnCommitUpdated`](#)
- [Evento `commentOnPullRequestCreated`](#)
- [Evento `commentOnPullRequestUpdated`](#)
- [Evento `pullRequestCreated`](#)
- [Evento `pullRequestSourceBranchUpdated`](#)
- [Evento `pullRequestStatusChanged`](#)
- [Evento `pullRequestMergeStatusUpdated`](#)
- [Evento `approvalRuleTemplateCreated`](#)

- [Evento approvalRuleTemplateUpdated](#)
- [Evento approvalRuleTemplateDeleted](#)
- [Evento approvalRuleTemplateAssociatedWithRepository](#)
- [Evento approvalRuleTemplateDisassociatedWithRepository](#)
- [Evento approvalRuleTemplateBatchAssociatedWithRepositories](#)
- [Evento approvalRuleTemplateBatchDisassociatedFromRepositories](#)
- [Evento pullRequestApprovalRuleCreated](#)
- [Evento pullRequestApprovalRuleDeleted](#)
- [Evento pullRequestApprovalRuleOverridden](#)
- [Evento pullRequestApprovalStateChanged](#)
- [Evento pullRequestApprovalRuleUpdated](#)
- [Evento reactionCreated](#)
- [Evento reactionUpdated](#)

Evento referenceCreated

En este evento de ejemplo, se ha creado una ramificación denominada myBranch en un repositorio denominado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "referenceCreated",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
    "referenceType": "branch",
    "referenceName": "myBranch",
    "referenceFullName": "refs/heads/myBranch",
```

```
    "commitId": "3e5983DESTINATION"
  }
}
```

Evento referenceUpdated

En este evento de ejemplo, una ramificación denominada myBranch se ha actualizado mediante una combinación en un repositorio denominado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "referenceUpdated",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
    "referenceType": "branch",
    "referenceName": "myBranch",
    "referenceFullName": "refs/heads/myBranch",
    "commitId": "7f0103fMERGE",
    "oldCommitId": "3e5983DESTINATION",
    "baseCommitId": "3e5a9bf1BASE",
    "sourceCommitId": "26a8f2SOURCE",
    "destinationCommitId": "3e5983DESTINATION",
    "mergeOption": "THREE_WAY_MERGE",
    "conflictDetailsLevel": "LINE_LEVEL",
    "conflictResolutionStrategy": "AUTOMERGE"
  }
}
```

Evento referenceDeleted

En este evento de ejemplo, se ha eliminado una ramificación denominada myBranch en un repositorio denominado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "referenceDeleted",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
    "referenceType": "branch",
    "referenceName": "myBranch",
    "referenceFullName": "refs/heads/myBranch",
    "oldCommitId": "26a8f2EXAMPLE"
  }
}
```

Evento unreferencedMergeCommitCreated

En este evento de ejemplo, se ha creado una confirmación de combinación sin referencia en un repositorio denominado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "unreferencedMergeCommitCreated",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",

```

```

    "commitId": "7f0103fMERGE",
    "baseCommitId": "3e5a9bf1BASE",
    "sourceCommitId": "26a8f2SOURCE",
    "destinationCommitId": "3e5983DESTINATION",
    "mergeOption": "SQUASH_MERGE",
    "conflictDetailsLevel": "LINE_LEVEL",
    "conflictResolutionStrategy": "AUTOMERGE"
  }
}

```

Evento commentOnCommitCreated

En este evento de ejemplo, un usuario federado llamado `Mary_Major` comentó en una confirmación. En este ejemplo, su proveedor de identidades federadas configuró etiquetas de sesión para `displayName` y `emailAddress`. Esa información se incluye en el evento.

```

{
  "version": "0",
  "id": "e9dce2e9-EXAMPLE",
  "detail-type": "CodeCommit Comment on Commit",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-09-29T20:20:39Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "beforeCommitId": "3c5dEXAMPLE",
    "repositoryId": "7dd1EXAMPLE...",
    "inReplyTo": "695bEXAMPLE...",
    "notificationBody": "A comment event occurred in the following repository:
MyDemoRepo. The display name for the user is Mary Major. The email address for
the user is mary_major@example.com. The user arn:aws:sts::123456789012:federated-
user/Mary_Major made a comment. The comment was made on the following comment ID:
463bEXAMPLE.... For more information, go to the AWS CodeCommit console at https://us-
east-2.console.aws.amazon.com/codecommit/home?region=us-east-2#/repository/MyDemoRepo/
compare/3c5dEXAMPLE...f4d5EXAMPLE#463bEXAMPLE....",
    "commentId": "463bEXAMPLE...",
    "afterCommitId": "f4d5EXAMPLE",
    "event": "commentOnCommitCreated",
    "repositoryName": "MyDemoRepo",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Mary_Major",

```

```

    "displayName": "Mary Major",
    "emailAddress": "mary_major@example.com"
  }
}

```

Evento commentOnCommitUpdated

En este evento de ejemplo, un usuario que asumió un rol Admin con un nombre de sesión Mary_Major editó un comentario en una confirmación. En este ejemplo, el rol incluía etiquetas de sesión configuradas para `displayName` y `emailAddress`. Esa información se incluye en el evento.

```

{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Commit",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "afterCommitId": "53812581",
    "beforeCommitId": "03314446",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "commentId": "a7e5471e-EXAMPLE",
    "event": "commentOnCommitUpdated",
    "inReplyTo": "bdb07d47-EXAMPLE",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Mary
Major. The email address for the user is mary_major@example.com. The user
arn:aws:sts::123456789012:federated-user/Mary_Major updated a comment or
replied to a comment. The comment was made on the following comment ID:
bdb07d47-6fe9-47b0-a839-b93cc743b2ac:468cd1cb-2dfb-4f68-9636-8de52431d1d6.
For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/
compare/0331444646178429589969823096709582251768/.../5381258150293783361471680277136017291382?
region\u003dus-east-2",
    "repositoryId": "12345678-1234-1234-1234-123456789012",
    "repositoryName": "MyDemoRepo",
    "displayName": "Mary Major",
    "emailAddress": "mary_major@example.com"
  }
}

```

```
}
}
```

Evento commentOnPullRequestCreated

En este evento de ejemplo, un usuario federado llamado Saanvi_Sarkar comentó en una solicitud de extracción. En este ejemplo, su proveedor de identidades federadas configuró etiquetas de sesión para `displayName` y `emailAddress`. Esa información se incluye en el evento.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Pull Request",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "beforeCommitId": "3c5dEXAMPLE",
    "repositoryId": "7dd1EXAMPLE...",
    "inReplyTo": "695bEXAMPLE...",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Saanvi
Sarkar. The email address for the user is saanvi_sarkar@example.com. The user
arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar made a comment. The comment
was made on the following Pull Request: 201. For more information, go to the AWS
CodeCommit console https://us-east-2.console.aws.amazon.com/codecommit/home?region=us-
east-2#/repository/MyDemoRepo/pull-request/201/activity#3276EXAMPLE...",
    "commentId": "463bEXAMPLE...",
    "afterCommitId": "f4d5EXAMPLE",
    "event": "commentOnPullRequestCreated",
    "repositoryName": "MyDemoRepo",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar",
    "pullRequestId": "201",
    "displayName": "Saanvi Sarkar",
    "emailAddress": "saanvi_sarkar@example.com"
  }
}
```


Evento commentOnPullRequestUpdated

En este evento de ejemplo, un usuario federado llamado Saanvi_Sarkar editó un comentario en una solicitud de extracción. En este ejemplo, su proveedor de identidades federadas configuró etiquetas de sesión para `displayName` y `emailAddress`. Esa información se incluye en el evento.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Pull Request",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "afterCommitId": "96814774EXAMPLE",
    "beforeCommitId": "6031971EXAMPLE",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar",
    "commentId": "40cb52f0-EXAMPLE",
    "event": "commentOnPullRequestUpdated",
    "inReplyTo": "1285e713-EXAMPLE",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Saanvi
Sarkar. The email address for the user is saanvi_sarkar@example.com. The user
arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar updated a comment or
replied to a comment. The comment was made on the following Pull Request:
1. For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/1/activity#40cb52f0-aac7-4c43-b771-601eff02EXAMPLE",
    "pullRequestId": "1",
    "repositoryId": "12345678-1234-1234-1234-123456789012",
    "repositoryName": "MyDemoRepo"
  }
}
```

Evento pullRequestCreated

En este evento de ejemplo, se creó una solicitud de extracción en un repositorio denominado MyDemoRepo por un usuario que asumió un rol Admin con un nombre de sesión de Mary_Major.

No se proporcionó información de etiqueta de sesión, por lo que la información no se incluye en el evento.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Feb 9 2019 10:18:42 PDT ",
    "description": "An example description.",
    "destinationCommit": "12241970EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestCreated",
    "isMerged": "False",
    "lastModifiedDate": "Tue Feb 9 2019 10:18:42 PDT",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. User: arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major. Event: Created. The pull request was created with the following information: Pull Request ID as 1 and title as My Example Pull Request. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1",
    "pullRequestId": "1",
    "pullRequestStatus": "Open",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9bEXAMPLE",
    "sourceCommit": "2774290EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

Evento pullRequestSourceBranchUpdated

En este evento de ejemplo, un usuario que asumió un rol denominado Admin con un nombre de sesión de Mary_Major actualizó la ramificación de origen denominada test-branch para una solicitud de extracción con el ID de 1.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Feb 9 2019 10:18:42 PDT",
    "description": "An example description.",
    "destinationCommit": "7644990EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestSourceBranchUpdated",
    "isMerged": "False",
    "lastModifiedDate": "Tue Feb 9 2019 10:18:42 PDT",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:sts::123456789012:assumed-role/
Admin/Mary_Major. Event: Updated. The user updated the following pull request:
1. The pull request was updated with one or more commits to the source branch:
test-branch. For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Open",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9b4EXAMPLE",
    "sourceCommit": "64875001EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

```
}
```

Evento pullRequestStatusChanged

En este evento de ejemplo, un usuario que asumió un rol denominado Admin con el nombre de sesión Mary_Major cerró una solicitud de extracción con el ID de 1. La solicitud de extracción no se ha combinado.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Jun 18 10:34:20 PDT 2019",
    "description": "An example description.",
    "destinationCommit": "95149731EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestStatusChanged",
    "isMerged": "False",
    "lastModifiedDate": "Tue Jun 18 10:34:20 PDT 2019",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major updated the following PullRequest 1. The pull request status has been updated. The status is closed. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Closed",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9bEXAMPLE",
    "sourceCommit": "4409936EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

```
}
```

Evento pullRequestMergeStatusUpdated

En este evento de ejemplo, un usuario que asumió un rol denominado Admin con un nombre de sesión Mary_Major combinó una solicitud de extracción con el ID de 1.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Mon Mar 11 14:42:31 PDT 2019",
    "description": "An example description.",
    "destinationCommit": "4376719EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestMergeStatusUpdated",
    "isMerged": "True",
    "lastModifiedDate": "Mon Mar 11 14:42:31 PDT 2019",
    "mergeOption": "FAST_FORWARD_MERGE",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major updated the following PullRequest 1. The pull request merge status has been updated. The status is merged. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Closed",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9beEXAMPLE",
    "sourceCommit": "0701696EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

```
}
```

Evento approvalRuleTemplateCreated

En este evento de ejemplo, un usuario con un nombre de usuario de IAM `Mary_Major` creó una plantilla de regla de aprobación denominada `2-approvers-required-for-main`.

```
{
  "version": "0",
  "id": "f7702227-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateCreated",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template with the following name has been
created: 2-approvers-required-for-main. The ID of the created template is: d7385967-
EXAMPLE. For more information, go to the AWS CodeCommit console.",
    "repositories": {}
  }
}
```

Evento approvalRuleTemplateUpdated

En este evento de ejemplo, un usuario con un nombre de usuario de IAM `Mary_Major` editó una plantilla de regla de aprobación denominada `2-approvers-required-for-main`. La plantilla de regla de aprobación no está asociada a ningún repositorio.

```
{
  "version": "0",
  "id": "66403118-EXAMPLE",
```

```

"detail-type": "CodeCommit Approval Rule Template Change",
"source": "aws.codecommit",
"account": "123456789012",
"time": "2019-11-12T23:03:30Z",
"region": "us-east-2",
"resources": [

],
"detail": {
  "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
  "approvalRuleTemplateId": "c9d2b844-EXAMPLE",
  "approvalRuleTemplateName": "2-approvers-required-for-main",
  "callerUserArn": "arn:aws:iam::123456789012:user\Mary_Major",
  "creationDate": "Tue Nov 12 23:03:06 UTC 2019",
  "event": "approvalRuleTemplateDeleted",
  "lastModifiedDate": "Tue Nov 12 23:03:20 UTC 2019",
  "notificationBody": "A approval rule template event occurred in the following AWS
CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template with the following name has been
deleted: 2-approvers-required-for-main. The ID of the updated template is: c9d2b844-
EXAMPLE. For more information, go to the AWS CodeCommit console.",
  "repositories": {}
}
}

```

Evento approvalRuleTemplateDeleted

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Majoreliminó una plantilla de regla de aprobación denominada 2-approvers-required-for-main. La plantilla de regla de aprobación no está asociada a ningún repositorio.

```

{
  "version": "0",
  "id": "66403118-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:03:30Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "approvalRuleTemplateContentSha256": "4f3de6632EXAMPLE",
    "approvalRuleTemplateId": "c9d2b844-EXAMPLE",

```

```

    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user\Mary_Major",
    "creationDate": "Tue Nov 12 23:03:06 UTC 2019",
    "event": "approvalRuleTemplateUpdated",
    "lastModifiedDate": "Tue Nov 12 23:03:20 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following AWS
CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template with the following name has
been updated: 2-approvers-required-for-main. The ID of the updated template is:
c9d2b844-EXAMPLE. The after rule template content SHA256 is 4f3de663EXAMPLE. For more
information, go to the AWS CodeCommit console.",
    "repositories": {}
  }
}

```

Evento approvalRuleTemplateAssociatedWithRepository

En este evento de ejemplo, un usuario con un nombre de usuario de IAM `Mary_Major` asociado a una plantilla de regla de aprobación denominada `2-approvers-required-for-main` con un repositorio denominado `MyDemoRepo`.

```

{
  "version": "0",
  "id": "ea1c6d73-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateAssociatedWithRepository",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been associated with the

```



```

following repository: [MyDemoRepo]. For more information, go to the AWS CodeCommit
console.",
  "repositories": {
    "MyDemoRepo": "92ca7bf2-d878-49ed-a994-336a6cc7c574"
  }
}
}

```

Evento approvalRuleTemplateDisassociatedWithRepository

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Majordesvinculó una plantilla de regla de aprobación denominada 2-approvers-required-for-main de un repositorio denominado MyDemoRepo.

```

{
  "version": "0",
  "id": "ea1c6d73-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateDisassociatedFromRepository",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been disassociated from the
following repository: [MyDemoRepo]. For more information, go to the AWS CodeCommit
console.",
    "repositories": {
      "MyDemoRepo": "92ca7bf2-d878-49ed-a994-336a6cc7c574"
    }
  }
}

```

```
}
```

Evento approvalRuleTemplateBatchAssociatedWithRepositories

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major asoció en lote una plantilla de regla de aprobación denominada 2-approvers-required-for-main con un repositorio denominado MyDemoRepo y un repositorio denominado MyTestRepo.

```
{
  "version": "0",
  "id": "0f861e5b-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:39:09Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "c71c1fe0-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Tue Nov 12 23:38:57 UTC 2019",
    "event": "batchAssociateApprovalRuleTemplateWithRepositories",
    "lastModifiedDate": "Tue Nov 12 23:38:57 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template has been batch associated with the
following repository names: [MyDemoRepo, MyTestRepo]. For more information, go to the
AWS CodeCommit console.",
    "repositories": {
      "MyDemoRepo": "MyTestRepo"
    }
  }
}
```

Evento approvalRuleTemplateBatchDisassociatedFromRepositories

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Majordesvinculó en lote una plantilla de regla de aprobación denominada 2-approvers-required-for-main de un repositorio denominado MyDemoRepo y un repositorio denominado MyTestRepo.

```
{
  "version": "0",
  "id": "e08fc996-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:39:09Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "c71c1fe0-ff91-4db4-9a45-a86a7b6c474f",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Tue Nov 12 23:38:57 UTC 2019",
    "event": "batchDisassociateApprovalRuleTemplateFromRepositories",
    "lastModifiedDate": "Tue Nov 12 23:38:57 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been batch disassociated from
the following repository names: [MyDemoRepo, MyTestRepo]. For more information, go to
the AWS CodeCommit console.",
    "repositories": {
      "MyDemoRepo": "MyTestRepo"
    }
  }
}
```

Evento pullRequestApprovalRuleCreated

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major creó una regla de aprobación denominada 1-approver-needed para una solicitud de extracción con el identificador de 227.

```

{
  "version": "0",
  "id": "ad860f12-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:12:19Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleContentSha256": "f742eebbEXAMPLE",
    "approvalRuleId": "0a9b5dfc-EXAMPLE",
    "approvalRuleName": "1-approver-needed",
    "author": "arn:aws:iam::123456789012:user/Mary_Major",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleCreated",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Updated. Pull request: 227. Additional information: An approval rule has been
created with the following name: 1-approver-needed. For more information, go to the
AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/
repositories/MyDemoRepo/pull-requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
      "MyDemoRepo"
    ],
    "revisionId": "3b8cecab3EXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
  }
}

```

Evento pullRequestApprovalRuleDeleted

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major eliminó una regla de aprobación denominada 1-approver-needed para una solicitud de extracción con el identificador de 227. Un usuario de IAM llamado Saanvi_Sarkar creó originalmente la regla de aprobación.

```
{
  "version": "0",
  "id": "c1c3509d-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:12:19Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleContentSha256": "f742eebbEXAMPLE",
    "approvalRuleId": "0a9b5dfc-EXAMPLE",
    "approvalRuleName": "1-approver-needed",
    "author": "arn:aws:iam::123456789012:user/Saanvi_Sarkar",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleDeleted",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Created. Pull request: 227. Additional information: An approval rule has been
deleted: 1-approver-needed was deleted. For more information, go to the AWS CodeCommit
console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/
MyDemoRepo/pull-requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
      "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
```

```

    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
  }
}

```

Evento pullRequestApprovalRuleOverridden

En este evento de ejemplo, los requisitos de regla de aprobación para una solicitud de extracción los ha apartado (ANULAR) un usuario con un nombre de usuario de IAM Mary_Major. La solicitud de extracción fue creada por un usuario de IAM llamado Li_Juan.

```

{
  "version": "0",
  "id": "52d2cb73-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:12:19Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:iam::123456789012:user/Li_Juan",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleOverridden",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Updated. Pull request name: 227. Additional information: An override
event has occurred for the approval rules for this pull request. Override status:
OVERRIDE. For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/227?region=us-east-2",
    "overrideStatus": "OVERRIDE",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",

```

```

    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

En este evento de ejemplo, se han restablecido los requisitos de regla de aprobación para una solicitud de extracción (REVOKE).

```

{
    "version": "0",
    "id": "2895482d-13eb-b783-270d-76588e6029fa",
    "detail-type": "CodeCommit Pull Request State Change",
    "source": "aws.codecommit",
    "account": "123456789012",
    "time": "2019-11-06T19:12:19Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
    ],
    "detail": {
        "author": "arn:aws:iam::123456789012:user/Li_Juan",
        "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
        "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
        "description": "An An example description.",
        "destinationCommit": "194fdf00EXAMPLE",
        "destinationReference": "refs/heads/main",
        "event": "pullRequestApprovalRuleOverridden",
        "isMerged": "False",
        "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
        "notificationBody": "A pull request event occurred in the following
AWS CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/
Mary_Major. Event: Updated. Pull request name: 227. Additional information: An
override event has occurred for the approval rules for this pull request. Override
status: REVOKE. For more information, go to the AWS CodeCommit console https://
us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/227?region=us-east-2",
        "overrideStatus": "REVOKE",
        "pullRequestId": "227",
    }
}

```

```

    "pullRequestStatus": "Open",
    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

Evento pullRequestApprovalStateChanged

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major ha aprobado una solicitud de extracción.

```

{
    "version": "0",
    "id": "53e5d7e9-986c-1ebf-9d8b-ebef5596da0e",
    "detail-type": "CodeCommit Pull Request State Change",
    "source": "aws.codecommit",
    "account": "123456789012",
    "time": "2019-11-06T19:12:19Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
    ],
    "detail": {
        "approvalStatus": "APPROVE",
        "author": "arn:aws:iam::123456789012:user/Li_Juan",
        "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
        "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
        "description": "An An example description.",
        "destinationCommit": "194fdf00EXAMPLE",
        "destinationReference": "refs/heads/main",
        "event": "pullRequestApprovalStateChanged",
        "isMerged": "False",
        "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
        "notificationBody": "A pull request event occurred in the following
AWS CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/
Mary_Major. Event: Updated. Pull request name: 227. Additional information:
A user has changed their approval state for the pull request. State change:
APPROVE. For more information, go to the AWS CodeCommit console https://us-

```



```

east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major ha revocado una aprobación de solicitud de extracción.

```

{
    "version": "0",
    "id": "25e183d7-d01a-4e07-2bd9-b2d56ebecc81",
    "detail-type": "CodeCommit Pull Request State Change",
    "source": "aws.codecommit",
    "account": "123456789012",
    "time": "2019-11-06T19:12:19Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
    ],
    "detail": {
        "approvalStatus": "REVOKE",
        "author": "arn:aws:iam::123456789012:user/Li_Juan",
        "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
        "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
        "description": "An An example description.",
        "destinationCommit": "194fdf00EXAMPLE",
        "destinationReference": "refs/heads/main",
        "event": "pullRequestApprovalStateChanged",
        "isMerged": "False",
        "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
        "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Updated. Pull request name: 227. Additional information: A user has changed
their approval state for the pull request. State change: REVOKE. For more information,

```

```

go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/
codecommit/repositories/MyDemoRepo/pull-requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

Evento pullRequestApprovalRuleUpdated

En este evento de ejemplo, un usuario con un nombre de usuario de IAM Mary_Major ha editado una regla de aprobación para una solicitud de extracción. También es el usuario que autorizó la solicitud de extracción.

```

{
    "version": "0",
    "id": "21b1c819-2889-3528-1cb8-3861aacf9d42",
    "detail-type": "CodeCommit Pull Request State Change",
    "source": "aws.codecommit",
    "account": "123456789012",
    "time": "2019-11-06T19:12:19Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
    ],
    "detail": {
        "approvalRuleContentSha256": "f742eebbEXAMPLE",
        "approvalRuleId": "0a9b5dfc-EXAMPLE",
        "approvalRuleName": "1-approver-needed",
        "author": "arn:aws:iam::123456789012:user/Mary_Major",
        "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
        "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
        "description": "An example description.",
        "destinationCommit": "194fdf00EXAMPLE",
        "destinationReference": "refs/heads/main",
        "event": "pullRequestApprovalRuleUpdated",
        "isMerged": "False",
    }
}

```

```

    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following
AWS CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/
Mary_Major. Event: Updated. Pull request name: 227. The content of an approval
rule has been updated for the pull request. The name of the updated rule is: 1-
approver-needed. For more information, go to the AWS CodeCommit console https://
us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecab3EXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

Evento reactionCreated

En este caso de ejemplo, un usuario ha añadido una reacción a un comentario con un nombre de usuario de IAM de Mary_Major.

```

{
  "version":"0",
  "id":"59fccccd8-217a-32ce-2b05-561ed68a1c42",
  "detail-type":"CodeCommit Comment Reaction Change",
  "source":"aws.codecommit",
  "account":"123456789012",
  "time":"2020-04-14T00:49:03Z",
  "region":"us-east-2",
  "resources":[
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail":{
    "callerUserArn":"arn:aws:iam::123456789012:user/Mary_Major",
    "commentId":"28930161-EXAMPLE",
    "event":"commentReactionCreated",
    "notificationBody":"A comment reaction event occurred in the following AWS
CodeCommit Repository: MyDemoRepo. The user: arn:aws:iam::123456789012:user/Mary_Major
made a comment reaction # to the comment with comment ID: 28930161-EXAMPLE",

```

```

    "reactionEmojis":["#"],
    "reactionShortcodes":[":thumbsdown:"],
    "reactionUnicode":["U+1F44E"],
    "repositoryId":"12345678-1234-5678-abcd-12345678abcd",
    "repositoryName":"MyDemoRepo"
  }
}

```

Evento reactionUpdated

En este caso de ejemplo, un usuario ha actualizado una reacción a un comentario con un nombre de usuario de IAM de Mary_Major. Los usuarios solo pueden actualizar sus propias reacciones.

```

{
  "version":"0",
  "id":"0844ed99-a53f-3bdb-6048-4de315516889",
  "detail-type":"CodeCommit Comment Reaction Change",
  "source":"aws.codecommit",
  "account":"123456789012",
  "time":"2020-04-22T23:19:42Z",
  "region":"us-east-2",
  "resources":[
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail":{
    "callerUserArn":"arn:aws:iam::123456789012:user/Mary_Major",
    "commentId":"28930161-EXAMPLE",
    "event":"commentReactionUpdated",
    "notificationBody":"A comment reaction event occurred in the following AWS
CodeCommit Repository: MyDemoRepo. The user: arn:aws:iam::123456789012:user/Mary_Major
updated a reaction :smile: to the comment with comment ID: 28930161-EXAMPLE",
    "reactionEmojis":[
      "#"
    ],
    "reactionShortcodes":[
      ":smile:"
    ],
    "reactionUnicode":["
      U+1F604
    "],
    "repositoryId":"12345678-1234-5678-abcd-12345678abcd",
    "repositoryName":"MyDemoRepo"
  }
}

```

```
}
```

Registrar llamadas a la AWS CodeCommit API con AWS CloudTrail

CodeCommit está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en CodeCommit. CloudTrail captura todas las llamadas a la API CodeCommit como eventos, incluidas las llamadas desde la CodeCommit consola, tu cliente Git y las llamadas de código a CodeCommit APIs. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para CodeCommit. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar el destinatario de la solicitud CodeCommit, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

CodeCommit información en CloudTrail

CloudTrail está activado en su cuenta de Amazon Web Services al crear la cuenta. Cuando se produce una actividad en CodeCommit, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de Amazon Web Services. Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos](#).

Para obtener un registro continuo de los eventos en su cuenta de Amazon Web Services, incluidos los eventos de CodeCommit, cree una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de . La ruta registra los eventos de todas las regiones de la AWS partición y entrega los archivos de registro al bucket de Amazon S3 que especifique. 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 Integraciones y servicios compatibles](#)
- [Configuración de las notificaciones de Amazon SNS para CloudTrail](#)

- [Recibir archivos de CloudTrail registro de varias regiones](#) y [recibir archivos de CloudTrail registro de varias cuentas](#)

Cuando se habilita el CloudTrail registro en su cuenta de Amazon Web Services, las llamadas a la API realizadas a CodeCommit las acciones se rastrean en los archivos de CloudTrail registro, donde se escriben junto con otros registros de AWS servicio. CloudTrail determina cuándo crear y escribir en un nuevo archivo en función del período de tiempo y del tamaño del archivo.

Todas CodeCommit las acciones se registran CloudTrail, incluidas algunas (por ejemplo `GetObjectIdentifier`) que actualmente no están documentadas en la [Referencia de la AWS CodeCommit API](#), sino que se denominan permisos de acceso y se documentan en ella [CodeCommit referencia de permisos](#). Por ejemplo, las llamadas a las acciones `ListRepositories` (en las AWS CLI, `aws codecommit list-repositories`), `CreateRepository` (`aws codecommit create-repository`) y `PutRepositoryTriggers` (`aws codecommit put-repository-triggers`) generan entradas en los archivos de CloudTrail registro, así como las llamadas del cliente Git a `GitPull` y `GitPush`. Además, si tienes un CodeCommit repositorio configurado como fuente para una canalización CodePipeline, verás las llamadas a las acciones de permiso de CodeCommit acceso, por ejemplo, `UploadArchive` desde CodePipeline. Como se CodeCommit utiliza AWS Key Management Service para cifrar y descifrar repositorios, también verás las llamadas `Encrypt` y `Decrypt` las acciones que se CodeCommit realicen desde los registros. AWS KMS CloudTrail

Cada entrada de registro contiene información sobre quién generó la solicitud. La información de identidad del usuario en la entrada de registro le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado o bien un usuario de rol asumido.
- Si la solicitud la realizó otro servicio AWS

Para obtener más información, consulte el [Elemento `userIdentity` de CloudTrail](#).

Puede almacenar sus archivos de registro en su bucket de Amazon S3 durante todo el tiempo que desee, pero también puede definir reglas de ciclo de vida de Amazon S3 para archivar o eliminar archivos de registro automáticamente. De forma predeterminada, los archivos de registro se cifran con el cifrado del servidor (SSE) de Amazon S3.

Descripción de las entradas de los archivos de CodeCommit registro

CloudTrail Los archivos de registro pueden contener una o más entradas de registro. Cada entrada muestra varios eventos con formato JSON. Un evento de registro representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Las entradas de registro no son un rastro de la pila ordenada de las llamadas API públicas, por lo que no aparecen en ningún orden específico.

Note

Este ejemplo se ha manipulado para mejorar la legibilidad. En un archivo de CloudTrail registro, todas las entradas y eventos se concatenan en una sola línea. Este ejemplo también se ha limitado a una sola entrada. CodeCommit En un archivo de CloudTrail registro real, puede ver las entradas y los eventos de varios AWS servicios.

Contenido

- [Ejemplo: una entrada de registro para enumerar CodeCommit repositorios](#)
- [Ejemplo: una entrada de registro para crear un repositorio CodeCommit](#)
- [Ejemplos: entradas de registro para llamadas de extracción de Git a un CodeCommit repositorio](#)
- [Ejemplo: una entrada de registro para una inserción correcta en un repositorio CodeCommit](#)

Ejemplo: una entrada de registro para enumerar CodeCommit repositorios

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `ListRepositories` acción.

Note

Aunque `ListRepositories` devuelve una lista de repositorios, las respuestas no mutables no se registran en los CloudTrail registros, por lo que `responseElements` se muestran como `null` en el archivo de registro.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
```

```

    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T17:57:36Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "ListRepositories",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "aws-cli/1.10.53 Python/2.7.9 Windows/8 botocore/1.4.43",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "cb8c167e-EXAMPLE",
  "eventID": "e3c6f4ce-EXAMPLE",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "apiVersion": "2015-04-13",
  "recipientAccountId": "444455556666"
}

```

Ejemplo: una entrada de registro para crear un repositorio CodeCommit

El siguiente ejemplo muestra una entrada de CloudTrail registro que muestra la `CreateRepository` acción en la región EE.UU. Este (Ohio).

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "CreateRepository",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",

```



```

"userAgent": "aws-cli/1.10.53 Python/2.7.9 Windows/8 boto3/1.4.43",
"requestParameters": {
  "repositoryDescription": "Creating a demonstration repository.",
  "repositoryName": "MyDemoRepo"
},
"responseElements": {
  "repositoryMetadata": {
    "arn": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "creationDate": "Dec 14, 2016 6:19:14 PM",
    "repositoryId": "8afe792d-EXAMPLE",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryName": "MyDemoRepo",
    "accountId": "111122223333",
    "cloneUrlHttp": "https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryDescription": "Creating a demonstration repository.",
    "lastModifiedDate": "Dec 14, 2016 6:19:14 PM"
  }
},
"requestID": "d148de46-EXAMPLE",
"eventID": "740f179d-EXAMPLE",
"readOnly": false,
"resources": [
  {
    "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "accountId": "111122223333",
    "type": "AWS::CodeCommit::Repository"
  }
],
"eventType": "AwsApiCall",
"apiVersion": "2015-04-13",
"recipientAccountId": "111122223333"
}

```

Ejemplos: entradas de registro para llamadas de extracción de Git a un CodeCommit repositorio

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que muestra la `GitPull` acción en el lugar en el que ya up-to-date se encuentra el repositorio local.

```

{
  "eventVersion": "1.05",

```

```

"userIdentity": {
  "type": "IAMUser",
  "principalId": "AIDACKCEVSQ6C2EXAMPLE",
  "arn": "arn:aws:iam::444455556666:user/Mary_Major",
  "accountId": "444455556666",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "userName": "Mary_Major"
},
"eventTime": "2016-12-14T18:19:15Z",
"eventSource": "codecommit.amazonaws.com",
"eventName": "GitPull",
"awsRegion": "us-east-2",
"sourceIPAddress": "203.0.113.12",
"userAgent": "git/2.11.0.windows.1",
"requestParameters": null,
"responseElements": null,
"additionalEventData": {
  "protocol": "HTTP",
  "dataTransferred": false,
  "repositoryName": "MyDemoRepo",
  "repositoryId": "8afe792d-EXAMPLE",
},
"requestID": "d148de46-EXAMPLE",
"eventID": "740f179d-EXAMPLE",
"readOnly": true,
"resources": [
  {
    "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "accountId": "111122223333",
    "type": "AWS::CodeCommit::Repository"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `GitPull` acción cuando el repositorio local no está presente up-to-date y, por lo tanto, los datos se transfieren del CodeCommit repositorio al repositorio local.

```

{
  "eventVersion": "1.05",
  "userIdentity": {

```

```
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "GitPull",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "git/2.10.1",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "protocol": "HTTP",
    "capabilities": [
      "multi_ack_detailed",
      "side-band-64k",
      "thin-pack"
    ],
    "dataTransferred": true,
    "repositoryName": "MyDemoRepo",
    "repositoryId": "8afe792d-EXAMPLE",
    "shallow": false
  },
  "requestID": "d148de46-EXAMPLE",
  "eventID": "740f179d-EXAMPLE",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "accountId": "111122223333",
      "type": "AWS::CodeCommit::Repository"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Ejemplo: una entrada de registro para una inserción correcta en un repositorio CodeCommit

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra que la `GitPush` acción se ha realizado correctamente. La acción `GitPush` aparece dos veces en una entrada de registro de un envío correcto.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "GitPush",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "git/2.10.1",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "protocol": "HTTP",
    "dataTransferred": false,
    "repositoryName": "MyDemoRepo",
    "repositoryId": "8afe792d-EXAMPLE",
  },
  "requestID": "d148de46-EXAMPLE",
  "eventID": "740f179d-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "accountId": "111122223333",
      "type": "AWS::CodeCommit::Repository"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
```

```
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "GitPush",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "git/2.10.1",
  "requestParameters": {
    "references": [
      {
        "commit": "100644EXAMPLE",
        "ref": "refs/heads/main"
      }
    ]
  },
  "responseElements": null,
  "additionalEventData": {
    "protocol": "HTTP",
    "capabilities": [
      "report-status",
      "side-band-64k"
    ],
    "dataTransferred": true,
    "repositoryName": "MyDemoRepo",
    "repositoryId": "8afe792d-EXAMPLE",
  },
  "requestID": "d148de46-EXAMPLE",
  "eventID": "740f179d-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "accountId": "111122223333",
      "type": "AWS::CodeCommit::Repository"
    }
  ]
}
```

```
    }  
  ],  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "111122223333"  
}
```

Creación de CodeCommit recursos con AWS CloudFormation

AWS CodeCommit está integrado con AWS CloudFormation un servicio que le ayuda a modelar y configurar sus AWS recursos para que pueda dedicar menos tiempo a crear y administrar sus recursos e infraestructura. Crea una plantilla que describe todos los AWS recursos que desea (como los repositorios) y CloudFormation aprovisiona y configura esos recursos por usted.

Cuando la utilice CloudFormation, podrá reutilizar la plantilla para configurar los CodeCommit recursos de forma coherente y repetida. Describa sus recursos una vez y, a continuación, aprovisiona los mismos recursos una y otra vez en varias Cuentas de AWS regiones.

CodeCommit y CloudFormation plantillas

Para aprovisionar y configurar recursos CodeCommit y servicios relacionados, debe conocer [CloudFormation las plantillas](#). Las plantillas son archivos de texto con formato JSON o YAML. Estas plantillas describen los recursos que desea aprovisionar en sus CloudFormation pilas. Si no estás familiarizado con JSON o YAML, puedes usar CloudFormation Designer para ayudarte a empezar con CloudFormation las plantillas. Para obtener más información, consulte [¿Qué es Designer de CloudFormation ?](#) en la Guía del usuario de AWS CloudFormation .

CodeCommit admite la creación de repositorios en CloudFormation A diferencia de la creación de repositorios desde la consola o la línea de comandos, puede utilizarlos CloudFormation para crear repositorios y enviar automáticamente el código al repositorio recién creado desde un archivo.zip especificado en un bucket de Amazon S3. Para obtener más información, incluidos ejemplos de plantillas JSON y YAML para repositorios, consulte. [AWS::CodeCommit::Repository](#)

Al crear un CodeCommit repositorio mediante CloudFormation, tiene la opción de enviar código a ese repositorio como parte del proceso de creación, siempre que el archivo ocupe menos de 20 MB, configurando las propiedades en [AWS:CodeCommit: Repository Code](#). Puede especificar el bucket de Amazon S3 en el que se almacena el código y, si lo desea, utilizar la [BranchName propiedad](#) para especificar el nombre de la rama predeterminada que se creará en la confirmación inicial de ese código. Estas propiedades solo se utilizan en la creación inicial del repositorio y se ignoran en las actualizaciones de la pila. No puede usar estas propiedades para realizar confirmaciones adicionales en un repositorio ni para cambiar el nombre de la ramificación predeterminada después de realizar la confirmación inicial.

 Note

El 19 de enero de 2021, AWS cambió el nombre de la rama predeterminada CodeCommit de master a main. Este cambio de nombre afecta al comportamiento predeterminado CodeCommit al crear la confirmación inicial para los repositorios mediante la CodeCommit consola, el CodeCommit APIs AWS SDKs, el y el AWS CLI. Los repositorios creados con CloudFormation o AWS CDK con una confirmación inicial de código como parte de la creación se adaptarán a este cambio a partir del 4 de marzo de 2021. Este cambio no afecta a los repositorios o ramificaciones existentes. Los clientes que utilizan clientes Git locales para crear sus confirmaciones iniciales tienen un nombre de ramificación predeterminado que sigue la configuración de esos clientes de Git. Para obtener más información, consulte [Cómo trabajar con ramificaciones](#), [Crear una confirmación](#) y [Cambiar la configuración de una ramificación](#).

También puede crear plantillas que creen recursos relacionados, como [reglas de notificación](#) para repositorios, [proyectos de compilación de AWS CodeBuild](#), [aplicaciones de AWS CodeDeploy](#) y [canalizaciones de AWS CodePipeline](#).

Ejemplos de plantillas

En los siguientes ejemplos se crea un CodeCommit repositorio denominado *MyDemoRepo*. El repositorio recién creado se rellena con el código almacenado en un bucket de Amazon S3 denominado *MySourceCodeBucket* y colocado en una rama denominada *development*, que es la rama predeterminada del repositorio.

 Note

El nombre del bucket de Amazon S3 que contiene el archivo ZIP con el contenido que se confirmará en el nuevo repositorio se puede especificar usando un ARN o el nombre del bucket en la cuenta de Amazon Web Services. Se trata de una clave de objeto de Amazon S3, tal y como se define en la [Guía del desarrollador de Amazon S3](#).

JSON:

```
{
```



```

    "MyRepo": {
      "Type": "AWS::CodeCommit::Repository",
      "Properties": {
        "RepositoryName": "MyDemoRepo",
        "RepositoryDescription": "This is a repository for my project with code
from MySourceCodeBucket.",
        "Code": {
          "BranchName": "development",
          "S3": {
            "Bucket": "MySourceCodeBucket",
            "Key": "MyKey",
            "ObjectVersion": "1"
          }
        }
      }
    }
  }
}

```

YAML:

```

MyRepo:
  Type: AWS::CodeCommit::Repository
  Properties:
    RepositoryName: MyDemoRepo
    RepositoryDescription: This is a repository for my project with code from
MySourceCodeBucket.
  Code:
    BranchName: development
    S3:
      Bucket: MySourceCodeBucket,
      Key: MyKey,
      ObjectVersion: 1

```

Para obtener más ejemplos, consulte [AWS::CodeCommit::Repository](#).

CloudFormation, CodeCommit, y el AWS Cloud Development Kit (AWS CDK)

Los repositorios creados utilizando la CloudFormation funcionalidad de AWS CDK uso en su creación. Entender cómo funcionan CloudFormation las plantillas con CodeCommit los recursos puede ayudarte a crear y administrar tu AWS CDK código. Para obtener más información al respecto

AWS CDK, consulta la [Guía para AWS Cloud Development Kit \(AWS CDK\) desarrolladores](#) y la [Referencia de la AWS CDK API](#).

En el siguiente ejemplo AWS CDK de TypeScript se crea un CodeCommit repositorio denominado *MyDemoRepo*. El repositorio recién creado se rellena con el código almacenado en un bucket de Amazon S3 denominado *MySourceCodeBucket* y colocado en una rama denominada *development*, que es la rama predeterminada del repositorio.

```
import * as cdk from '@aws-cdk/core';
import codecommit = require('@aws-cdk/aws-codecommit');
export class CdkCodecommitStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);
    // The code creates a CodeCommit repository with a default branch name development
    new codecommit.CfnRepository(this, 'MyRepoResource', {
      repositoryName: "MyDemoRepo",
      code: {
        "branchName": "development",
        "s3": {
          "bucket": "MySourceCodeBucket",
          "key": "MyKey"
        }
      }
    },
  );
}
```

Obtenga más información sobre CloudFormation

Para obtener más información CloudFormation, consulte los siguientes recursos:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guía del usuario](#)
- [AWS CloudFormation Guía del usuario de la interfaz de línea de comandos](#)

Resolución de problemas de AWS CodeCommit

La siguiente información puede ayudarle a solucionar problemas comunes en AWS CodeCommit.

Temas

- [Solución de problemas de credenciales de Git y conexiones HTTPS a AWS CodeCommit](#)
- [Solución de problemas git-remote-codecommit y AWS CodeCommit](#)
- [Solución de problemas de conexiones SSH a AWS CodeCommit](#)
- [Solución de problemas del asistente de credenciales y las conexiones HTTPS a AWS CodeCommit](#)
- [Solución de problemas de clientes Git y AWS CodeCommit](#)
- [Solución de problemas de acceso y AWS CodeCommit](#)
- [Solución de problemas de configuración y AWS CodeCommit](#)
- [Solución de problemas de la consola y AWS CodeCommit](#)
- [Solución de problemas desencadenantes y AWS CodeCommit](#)
- [Activar la depuración](#)

Solución de problemas de credenciales de Git y conexiones HTTPS a AWS CodeCommit

La siguiente información puede ayudarte a solucionar problemas comunes al usar las credenciales de Git y HTTPS para conectarte a los AWS CodeCommit repositorios.

Temas

- [Credenciales de Git para AWS CodeCommit: sigo viendo una solicitud de credenciales cuando me conecto a mi CodeCommit repositorio en la terminal o en la línea de comandos](#)
- [Credenciales de Git para AWS CodeCommit: configuré las credenciales de Git, pero mi sistema no las usa](#)

Credenciales de Git para AWS CodeCommit: sigo viendo una solicitud de credenciales cuando me conecto a mi CodeCommit repositorio en la terminal o en la línea de comandos

Problema: cuando intentas insertar, extraer o interactuar con un CodeCommit repositorio desde el terminal o la línea de comandos, se te pide que proporciones un nombre de usuario y una contraseña, y debes proporcionar las credenciales de Git de tu usuario de IAM.

Soluciones posibles: las causas más comunes de este error son que el equipo local ejecute un sistema operativo que no admite la administración de credenciales, o que no tiene una utilidad de administración de credenciales instalado, o que no se han guardado las credenciales de Git para el usuario de IAM en uno de estos sistemas de gestión de credenciales. Según el sistema operativo y el entorno local, es posible que tenga que instalar un gestor de credenciales, configurar el gestor de credenciales que se incluye en el sistema operativo o personalizar su entorno local para utilizar el almacenamiento de credenciales. Por ejemplo, si su equipo ejecuta macOS, puede usar la utilidad Keychain Access para almacenar sus credenciales. Si su equipo utiliza Windows, puede utilizar el Administrador de credenciales de Git que se instala en Git para Windows. Para obtener más información, consulte las secciones [Para usuarios HTTPS mediante credenciales de Git](#) y [Almacenamiento de credenciales](#) en la documentación de Git.

Credenciales de Git para AWS CodeCommit: configuré las credenciales de Git, pero mi sistema no las usa

Problema: cuando intentas usarlo CodeCommit con un cliente de Git, parece que el cliente no usa las credenciales de Git para tu usuario de IAM.

Soluciones posibles: la causa más frecuente de este error es que el equipo se había configurado para utilizar la aplicación ayudante de credenciales que se incluye con la AWS CLI. Compruebe que las secciones de configuración del archivo .gitconfig sean similares a las siguientes y quítelas:

```
[credential "https://git-codecommit.*.amazonaws.com"]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

Guarde el archivo y, a continuación, abra una nueva sesión de línea de comandos o de terminal antes de intentar conectarse de nuevo.

También puede que tenga configurados varios ayudantes de credenciales o administradores en el equipo y su sistema puede tomar como valor predeterminado otra configuración. Para restablecer el ayudante de credenciales que se usa de forma predeterminada, puede utilizar la opción `--system` en lugar de `--global` o `--local` al ejecutar el comando `git config`.

Para obtener más información, consulte las secciones [Para usuarios HTTPS mediante credenciales de Git](#) y [Almacenamiento de credenciales](#) en la documentación de Git.

Solución de problemas git-remote-codecommit y AWS CodeCommit

La siguiente información puede ayudarle a solucionar problemas con git-remote-codecommit al conectarse a los repositorios de AWS CodeCommit .

Temas

- [Veo un error: git: 'remote-codecommit' no es un comando de git](#)
- [Aparece un error: error grave: no se ha podido encontrar el asistente remoto para 'codecommit'](#)
- [Error de clonación: no puedo clonar un CodeCommit repositorio desde un IDE](#)
- [Error de inserción o extracción: no puedo enviar o extraer confirmaciones de un IDE a un repositorio CodeCommit](#)
- [Error de instalación: aparece un externally-managed-environment error al intentar realizar la instalación git-remote-codecommit](#)

Veo un error: git: 'remote-codecommit' no es un comando de git

Problema: cuando intentas usarlo git-remote-codecommit, aparece un error que no git-remote-codecommit es un comando de git. Consulte 'git --help'».

Posibles soluciones: el motivo más común de este error es que no has añadido el git-remote-codecommit ejecutable a tu PATH o que la cadena contiene un error de sintaxis. Esto puede ocurrir cuando falta un guión entre git y remote-codecommit, o cuando se coloca un git adicional antes. git-remote-codecommit

También puede producirse este error si ha actualizado la instalación local a una versión no compatible. Para obtener más información sobre las versiones compatibles de Python, consulte [git-remote-codecommit](#).

Para obtener más información sobre la configuración y el uso git-remote-codecommit, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Aparece un error: error grave: no se ha podido encontrar el asistente remoto para 'codecommit'

Problema: Al intentar usarlo git-remote-codecommit, aparece un mensaje de error que indica «grave: no se ha podido encontrar el asistente remoto para 'codecommit'».

Soluciones posibles: las razones más comunes de este error son:

- La configuración no está completa para git-remote-codecommit
- La ha instalado git-remote-codecommit en una ubicación que no está en su ruta o que no está configurada como parte de la variable de Path entorno
- Python no está en su ruta o no está configurado como parte de la variable del entorno Path
- Está utilizando un terminal o una ventana de línea de comandos que no se ha reiniciado desde que se completó la instalación git-remote-codecommit

Para obtener más información sobre la configuración y el uso git-remote-codecommit, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Error de clonación: no puedo clonar un CodeCommit repositorio desde un IDE

Problema: al intentar clonar un CodeCommit repositorio en un IDE, aparece un error que indica que el punto final o la URL no son válidos.

Soluciones posibles: no todas IDEs admiten la URL utilizada git-remote-codecommit durante la clonación. Clone el repositorio localmente desde el terminal o la línea de comandos y, a continuación, agregue ese repositorio local a su IDE. Para obtener más información, consulte [Paso 3: Conectarse a la CodeCommit consola y clonar el repositorio](#).

Error de inserción o extracción: no puedo enviar o extraer confirmaciones de un IDE a un repositorio CodeCommit

Problema: cuando intenta extraer o insertar código en un IDE, aparece un error de conexión.

Posibles soluciones: la razón más común de este error es que el IDE no es compatible con el código auxiliar remoto de Git como `git-remote-codecommit`. En lugar de usar la funcionalidad del IDE para confirmar, insertar y extraer código, actualice el repositorio local manualmente desde la línea de comandos o el terminal usando comandos de Git.

Para obtener más información sobre el código auxiliar remoto y Git, consulte la [documentación de Git](#).

Error de instalación: aparece un externally-managed-environment error al intentar realizar la instalación git-remote-codecommit

Problema: Al intentar ejecutar el comando `pip install git-remote-codecommit`, aparece un error que indica que el entorno está gestionado externamente.

Posibles soluciones: El motivo más frecuente de este error es que está ejecutando una distribución (distro) de Python que define un archivo de marcadores EXTERNALLY-MANAGED. La mejor solución es crear y utilizar un entorno virtual.

Para obtener más información sobre los entornos administrados externamente y Python, consulte [Entornos administrados externamente](#) e [Instalación de paquetes en un entorno virtual mediante pip y venv](#) en la documentación de Python.

Solución de problemas de conexiones SSH a AWS CodeCommit

La siguiente información puede ayudarte a solucionar problemas comunes al usar SSH para conectarte a los repositorios. CodeCommit

Temas

- [Error de acceso: la clave pública se ha cargado correctamente en IAM pero se ha producido un error de conexión en los sistemas Linux, macOS o Unix](#)
- [Error de acceso: la clave pública se carga correctamente en IAM y SSH se prueba correctamente pero la conexión falla en los sistemas Windows](#)
- [Problema de autenticación: no se puede establecer la autenticidad del host al conectarse a un repositorio CodeCommit](#)
- [Error de IAM: "Formato no válido" al intentar añadir una clave pública en IAM](#)
- [Necesito acceder a los CodeCommit repositorios de varias cuentas de Amazon Web Services con credenciales SSH](#)

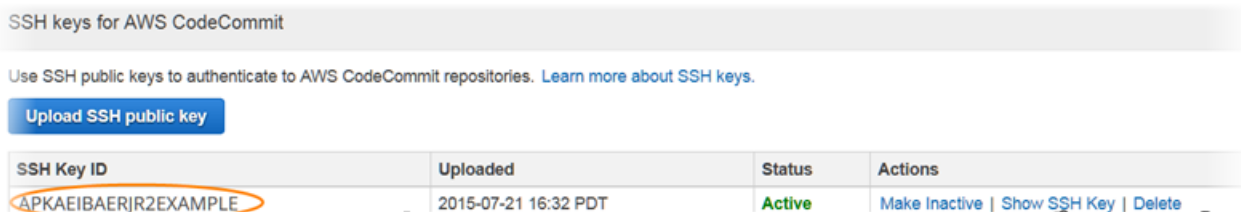
- [Git en Windows: el emulador de Bash o la línea de comandos se congela al intentar conectar mediante SSH](#)
- [El formato de clave pública requiere una especificación en algunas distribuciones de Linux](#)
- [Error de acceso: se denegó la clave pública SSH al conectarse a un repositorio CodeCommit](#)

Error de acceso: la clave pública se ha cargado correctamente en IAM pero se ha producido un error de conexión en los sistemas Linux, macOS o Unix

Problema: cuando intentas conectarte a un punto final de SSH para comunicarte con un CodeCommit repositorio, ya sea al probar la conexión o al clonar un repositorio, la conexión falla o se rechaza.

Soluciones posibles: el ID de la clave SSH asignado a su clave pública en IAM podría no estar asociado con su intento de conexión. [Es posible que no haya configurado un archivo de configuración](#), es posible que no tenga acceso al archivo de configuración, es posible que otro ajuste impida la lectura correcta del archivo de configuración, podría haber proporcionado el ID de la clave incorrecto, o es posible que haya proporcionado el ID del usuario de IAM en lugar del ID de la clave.

El ID de la clave de SSH puede encontrarse en el perfil de usuario de IAM de la consola de IAM:



SSH keys for AWS CodeCommit			
Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.			
Upload SSH public key			
SSH Key ID	Uploaded	Status	Actions
APKAEIBAERJR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make inactive Show SSH Key Delete

Note

Si has IDs cargado más de una clave SSH, las claves se muestran alfabéticamente por ID de clave, no por fecha de carga. Asegúrese de que ha copiado el ID de clave que está asociado a la fecha de carga correcta.

Intente conectarse con el siguiente comando:

```
ssh Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com
```


Si recibe un mensaje de éxito después de confirmar la conexión, significa que el ID de la clave de SSH es válido. Edite su archivo de configuración para asociar sus intentos de conexión a su clave pública en IAM. Si no desea editar el archivo de configuración, puede iniciar todos los intentos de conexión a su repositorio con su ID de la clave de SSH. Por ejemplo, si quisieras clonar un repositorio con un nombre *MyDemoRepo* sin modificar tu archivo de configuración para asociar tus intentos de conexión, ejecutarías el siguiente comando:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/
repos/MyDemoRepo my-demo-repo
```

Para obtener más información, consulte [Para conexiones SSH en Linux, macOS o Unix](#).

Error de acceso: la clave pública se carga correctamente en IAM y SSH se prueba correctamente pero la conexión falla en los sistemas Windows

Problema: cuando intentas usar un punto final SSH para clonar o comunicarte con un CodeCommit repositorio, aparece un mensaje de error que contiene la frase `No supported authentication methods available`.

Soluciones posibles: el motivo más común de este error es que tenga fijada una variable del entorno del sistema Windows que obliga a Windows a utilizar otro programa cuando intenta usar SSH. Por ejemplo, es posible que tenga que establecer una variable `GIT_SSH` para que apunte hacia un conjunto de herramientas de PuTTY (`plink.exe`). Puede tratarse de una configuración heredada o quizá sea necesario instalar uno o varios programas en el equipo. Si está seguro de que no necesita esta variable de entorno, puede eliminarla abriendo las propiedades del sistema.

Para solucionar este problema, abra un emulador de Bash y, a continuación, pruebe su conexión SSH de nuevo, pero esta vez incluyendo `GIT_SSH_COMMAND="SSH"` como prefijo. Por ejemplo, para clonar un repositorio mediante SSH:

```
GIT_SSH_COMMAND="ssh" git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo my-demo-repo
```

Podría surgir un problema similar si su versión de Windows requiere que incluya el ID de la clave de SSH en la cadena de conexión a la hora de conectarse mediante SSH en la línea de comandos de Windows. Pruebe la conexión de nuevo, pero ahora incluyendo el ID de la clave de SSH copiado de IAM en el comando. Por ejemplo:

```
git clone ssh://Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Problema de autenticación: no se puede establecer la autenticidad del host al conectarse a un repositorio CodeCommit

Problema: cuando intentas utilizar un punto final SSH para comunicarte con un CodeCommit repositorio, aparece un mensaje de advertencia que contiene la frase The authenticity of host '*host-name*' can't be established.

Soluciones posibles: puede que no haya configurado correctamente sus credenciales. Siga las instrucciones de [Para conexiones SSH en Linux, macOS o Unix](#) o [Para conexiones SSH en Windows](#).

Si has seguido estos pasos y el problema persiste, es posible que alguien esté intentando realizar un man-in-the-middle ataque. Cuando vea el siguiente mensaje, escriba no y, a continuación, pulse Intro.

```
Are you sure you want to continue connecting (yes/no)?
```

Asegúrese de que la huella digital y la clave pública de CodeCommit las conexiones coincidan con las documentadas en los temas de configuración de SSH antes de continuar con la conexión.

Huellas digitales públicas para CodeCommit

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.us-east-2.amazonaws.com	MD5	a9:6d:03:ed:08:42: 21:be:06:e1:e0:2a: d1:75:31:5e
git-codecommit.us-east-2.amazonaws.com	SHA256	31B1W2g5xn/NA2Ck6d yeJIrQ0Wvn7n8UEs56 fG6ZIZQ
git-codecommit.us-east-1.amazonaws.com	MD5	a6:9c:7d:bc:35:f5: d4:5f:8b:ba:6f:c8: bc:d4:83:84

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.us-east-1.amazonaws.com	SHA256	eLMY1j0DKA4uvDZc1/ KgtIayZANwX6t8+8is PtotBoY
git-codecommit.us-west-2.amazonaws.com	MD5	a8:68:53:e3:99:ac: 6e:d7:04:7e:f7:92: 95:77:a9:77
git-codecommit.us-west-2.amazonaws.com	SHA256	0pJx9SQpkbPUAHwy58 UVIq0IHcyo1fwCp00u VgcAWPo
git-codecommit.eu-west-1.amazonaws.com	MD5	93:42:36:ea:22:1f: f1:0f:20:02:4a:79: ff:ea:12:1d
git-codecommit.eu-west-1.amazonaws.com	SHA256	tKjRk0L8dmJyTmSbeS dN1S8F/f0iq13Rlvqg TOP1UyQ
git-codecommit.ap-northeast-1.amazonaws.com	MD5	8e:a3:f0:80:98:48: 1c:5c:6f:59:db:a7: 8f:6e:c6:cb
git-codecommit.ap-northeast-1.amazonaws.com	SHA256	Xk/WeYD/K/bnBybzhi uu4dWpBJtXPf7E30jH U7se40w
git-codecommit.ap-southeast-1.amazonaws.com	MD5	65:e5:27:c3:09:68: 0d:8e:b7:6d:94:25: 80:3e:93:cf
git-codecommit.ap-southeast-1.amazonaws.com	SHA256	ZIsVa70VzxrTIf+Rk4 UbhPv6Es22mSB3uTBo jfPXIno

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.ap-southeast-2.amazonaws.com	MD5	7b:d2:c1:24:e6:91: a5:7b:fa:c1:0c:35: 95:87:da:a0
git-codecommit.ap-southeast-2.amazonaws.com	SHA256	nYp+gHas80HY3DqbP4 yanCDFhqDVjseefVbH EXqH2Ec
git-codecommit.ap-southeast-3.amazonaws.com	MD5	64:d9:e0:53:19:4f: a8:91:9a:c3:53:22: a6:a8:ed:a6
git-codecommit.ap-southeast-3.amazonaws.com	SHA256	ATdkGSFhpqIu7RqUVT /1RZo6MLxxxUW9NoDV MbAc/6g
git-codecommit.me-central-1.amazonaws.com	MD5	bd:fa:e2:f9:05:84: d6:39:6f:bc:d6:8d: fe:de:61:76
git-codecommit.me-central-1.amazonaws.com	SHA256	grceUDWubo4MzG1Noa KZKUfrgPvfN3ijli0n Qr11TZA
git-codecommit.eu-central-1.amazonaws.com	MD5	74:5a:e8:02:fc:b2: 9c:06:10:b4:78:84: 65:94:22:2d
git-codecommit.eu-central-1.amazonaws.com	SHA256	MwGrkiEki8QkkBt1Ag XbYt0hoZYBnZF62VY5 RzGJEUy
git-codecommit.ap-northeast-2.amazonaws.com	MD5	9f:68:48:9b:5f:fc: 96:69:39:45:58:87: 95:b3:69:ed

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.ap-northeast-2.amazonaws.com	SHA256	eegAPQrWY9YsYo9ZHI K0mxetfXBHzAZd8Eya 53Qcwko
git-codecommit.sa-east-1.amazonaws.com	MD5	74:99:9d:ff:2b:ef: 63:c6:4b:b4:6a:7f: 62:c5:4b:51
git-codecommit.sa-east-1.amazonaws.com	SHA256	kW+VKB0jpRaG/ZbXkg btMQbKgEDK7JnISV3S VoyCmzU
git-codecommit.us-west-1.amazonaws.com	MD5	3b:76:18:83:13:2c: f8:eb:e9:a3:d0:51: 10:32:e7:d1
git-codecommit.us-west-1.amazonaws.com	SHA256	gzauWTWXDK2u5KuMMi 5vbKTmfyerdIwgSbzY B0DLpzg
git-codecommit.eu-west-2.amazonaws.com	MD5	a5:65:a6:b1:84:02: b1:95:43:f9:0e:de: dd:ed:61:d3
git-codecommit.eu-west-2.amazonaws.com	SHA256	r0Rwz5k/IHp/QyrRnf iM9j02D5UEqMbtFNTu DG2hNbs
git-codecommit.ap-south-1.amazonaws.com	MD5	da:41:1e:07:3b:9e: 76:a0:c5:1e:64:88: 03:69:86:21
git-codecommit.ap-south-1.amazonaws.com	SHA256	hUKwnTj7+Xpx4Kddb6 p45j4RazIJ4IhAMD8k 29it0fE

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.ap-south-2.amazonaws.com	MD5	bc:cc:9f:15:f8:f3: 58:a2:68:65:21:e2: 23:71:8d:ce
git-codecommit.ap-south-2.amazonaws.com	SHA256	Xe0CyZE0vgR5Xa2YUG qf+jn8/Ut7l7nX/Cms lSFNEig
git-codecommit.ca-central-1.amazonaws.com	MD5	9f:7c:a2:2f:8c:b5: 74:fd:ab:b7:e1:fd: af:46:ed:23
git-codecommit.ca-central-1.amazonaws.com	SHA256	Qz5puafQdANVprLlj6 r0Qyh4lCNsF6ob61dG cPtFS7w
git-codecommit.eu-west-3.amazonaws.com	MD5	1b:7f:97:dd:d7:76: 8a:32:2c:bd:2c:7b: 33:74:6a:76
git-codecommit.eu-west-3.amazonaws.com	SHA256	uw7c2FL564jVoFgtc+ ikzILnKBsZz7t9+CFd SJjKbLI
git-codecommit.us-gov-west-1.amazonaws.com	MD5	9f:6c:19:3b:88:cd: e8:88:1b:9c:98:6a: 95:31:8a:69
git-codecommit.us-gov-west-1.amazonaws.com	SHA256	djXQoSIFcg8vHe0KVH 1xW/g0F9X37tWTqu4H kng75x4
git-codecommit.us-gov-east-1.amazonaws.com	MD5	00:8d:b5:55:6f:05: 78:05:ed:ea:cb:3f: e6:f0:62:f2

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.us-gov-east-1.amazonaws.com	SHA256	fVb+R0z7qW7minenW+rUpAABRCRBTCzmETAJ EQrg98
git-codecommit.eu-north-1.amazonaws.com	MD5	8e:53:d8:59:35:88:82:fd:73:4b:60:8a:50:70:38:f4
git-codecommit.eu-north-1.amazonaws.com	SHA256	b6KSK7xKq+V8j17iuAcjqXsG7zkqoUZZmmhY YFBq1wQ
git-codecommit.me-south-1.amazonaws.com	MD5	0e:39:28:56:d5:41:e6:8d:fa:81:45:37:fb:f3:cd:f7
git-codecommit.me-south-1.amazonaws.com	SHA256	0+NTToCGgjHekiBu010ad7R0GEsz+DBLX0d/c9wc0JU
git-codecommit.ap-east-1.amazonaws.com	MD5	a8:00:3d:24:52:9d:61:0e:f6:e3:88:c8:96:01:1c:fe
git-codecommit.ap-east-1.amazonaws.com	SHA256	LafadYwUYW8h0NoTRpobjjNs9IRnbEwHtezD3aAIBX0
git-codecommit.cn-north-1.amazonaws.com.cn	MD5	11:7e:2d:74:9e:3b:94:a2:69:14:75:6f:5e:22:3b:b3
git-codecommit.cn-north-1.amazonaws.com.cn	SHA256	IYUXxH20pTDsyYMLIp+JY8CTLS4UX+ZC5JVZ XPRaxc8

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.cn-northwest-1.amazonaws.com.cn	MD5	2e:a7:fb:4c:33:ac: 6c:f9:aa:f2:bc:fb: 0a:7b:1e:b6
git-codecommit.cn-northwest-1.amazonaws.com.cn	SHA256	wqjd6eHd0+m0Bx+dCN uL0omUoCNjaDtZiEpW j5TmCfQ
git-codecommit.eu-south-1.amazonaws.com	MD5	b9:f6:5d:e2:48:92: 3f:a9:37:1e:c4:d0: 32:0e:fb:11
git-codecommit.eu-south-1.amazonaws.com	SHA256	1yXrWbCg3uQmJr11Xx B/ASR7ugW1Ysf5yzY0 JbudHsI
git-codecommit.ap-northeast-3.amazonaws.com	MD5	25:17:40:da:b9:d4: 18:c3:b6:b3:fb:ed: 1c:20:fe:29
git-codecommit.ap-northeast-3.amazonaws.com	SHA256	2B815B9F0AvwLnRxSV xUz4kDYmtEQUGGdQYP 80QLXhA
git-codecommit.af-south-1.amazonaws.com	MD5	21:a0:ba:d7:c1:d1: b5:39:98:8d:4d:7c: 96:f5:ca:29
git-codecommit.af-south-1.amazonaws.com	SHA256	C34ji3x/cnsDZjUpyN GXdE5pjHYimqJrQZ31 eTgqJHM
git-codecommit.il-central-1.amazonaws.com	MD5	04:74:89:16:98:7a: 61:b1:69:46:42:3c: d1:b4:ac:a9

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.il-central-1.amazonaws.com	SHA256	uFxhp51kUWh1eTLeYb xQVYm4RnNLNZ5Dbdm1 cgdS1/8

Error de IAM: "Formato no válido" al intentar añadir una clave pública en IAM

Problema: en IAM, al intentar configurar el uso de SSH con CodeCommit, aparece un mensaje de error con la frase «Invalid format cuando intentas añadir tu clave pública».

Posibles soluciones: IAM requiere que la clave pública esté codificada en formato ssh-rsa o PEM. Solo acepta claves públicas en formato OpenSSH y tiene requisitos adicionales, tal como se especifica [en la sección Uso de claves SSH en](#) la Guía del usuario de CodeCommit IAM. Si proporciona su clave pública en otro formato, o si la clave no contiene el número necesario de bits, aparece este error.

- Cuando copió la clave pública SSH, es posible que su sistema operativo haya introducido saltos de línea. Asegúrese de que no haya saltos de línea en la clave pública que agregue a IAM.
- Algunos sistemas operativos Windows no utilizan el formato OpenSSH. Para generar un par de claves y copiar el formato de OpenSSH exigido por IAM, consulte [the section called "Paso 3: Configura las claves públicas y privadas para Git y CodeCommit"](#).

Para obtener más información sobre los requisitos de las claves SSH en IAM, consulte [Uso de claves SSH en la Guía del usuario](#) de IAM. CodeCommit

Necesito acceder a los CodeCommit repositorios de varias cuentas de Amazon Web Services con credenciales SSH

Problema: quiero configurar el acceso SSH a CodeCommit los repositorios de más de una cuenta de Amazon Web Services.

Posibles soluciones: puede crear pares de public/private claves SSH únicos para cada cuenta de Amazon Web Services y configurar IAM con cada clave. A continuación, puede configurar su archivo ~/.ssh/config con información sobre cada usuario de IAM asociado a la clave pública. Por ejemplo:

```
Host codecommit-1
  Hostname git-codecommit.us-east-1.amazonaws.com
  User SSH-KEY-ID-1 # This is the SSH Key ID you copied from IAM in Amazon Web
  Services account 1 (for example, APKAEIBAERJR2EXAMPLE1).
  IdentityFile ~/.ssh/codecommit_rsa # This is the path to the associated public key
  file, such as id_rsa. We advise creating CodeCommit specific _rsa files.

Host codecommit-2
  Hostname git-codecommit.us-east-1.amazonaws.com
  User SSH-KEY-ID-2 # This is the SSH Key ID you copied from IAM in Amazon Web
  Services account 2 (for example, APKAEIBAERJR2EXAMPLE2).
  IdentityFile ~/.ssh/codecommit_2_rsa # This is the path to the other associated
  public key file. We advise creating CodeCommit specific _rsa files.
```

En esta configuración, podrá sustituir 'git-codecommit.us-east-1.amazonaws.com' por 'codecommit-2'. Por ejemplo, para clonar un repositorio en su segunda cuenta de Amazon Web Services:

```
git clone ssh://codecommit-2/v1/repos/YourRepositoryName
```

Para configurar un control remoto para su repositorio, ejecute git remote add. Por ejemplo:

```
git remote add origin ssh://codecommit-2/v1/repos/YourRepositoryName
```

Para ver más ejemplos, consulte [esta publicación del foro](#) y [esta contribución](#) en GitHub

Git en Windows: el emulador de Bash o la línea de comandos se congela al intentar conectar mediante SSH

Problema: después de configurar el acceso de SSH para Windows y confirmar la conectividad a la línea de comandos o el terminal, aparece un mensaje en que se indica que la clave de host del servidor no está almacenada en caché en el registro y que la solicitud para almacenar la clave en la memoria caché está congelada (no acepta ninguna entrada y/n/return) cuando intenta usar comandos, como git pull, git push o git clone en el símbolo del sistema o el emulador de Bash.

Soluciones posibles: la causa más común de este error es que su entorno de Git se ha configurado para utilizar algo distinto de OpenSSH para la autenticación (probablemente PuTTY). Se han notificado problemas con el almacenamiento en caché de claves en algunas configuraciones. Para solucionar este problema, pruebe uno de los siguientes pasos:

- Abra un emulador de Bash y añada el parámetro `GIT_SSH_COMMAND="ssh"` delante del comando de Git. Por ejemplo, si intenta enviar a un repositorio, en lugar de escribir `git push`, escriba:

```
GIT_SSH_COMMAND="ssh" git push
```

- Si tiene PuTTY instalado, abra PuTTY y, en Nombre de host (o dirección IP), introduzca el CodeCommit punto final al que desea acceder (por ejemplo, `git-codecommit.us-east-2.amazonaws.com`). Elija Open. Cuando PuTTY Security Alert se lo solicite, elija Yes (Sí) para almacenar la clave en caché de forma permanente.
- Renombre o elimine la variable de entorno `GIT_SSH` si ya no la utiliza. A continuación, abra una nueva sesión del símbolo del sistema o del emulador de Bash y pruebe el comando de nuevo.

Para otras soluciones, consulta [Git se congela clone/pull continuamente en Store key in cache en Stack Overflow](#).

El formato de clave pública requiere una especificación en algunas distribuciones de Linux

Problema: al intentar configurar un key pair público-privado, aparece un error.

Posibles soluciones: algunas distribuciones de Linux requieren una línea de configuración adicional en el archivo `~/.ssh/config` que especifique los tipos de claves públicas aceptados. Para obtener más información, consulte la documentación para su distribución sobre `PubkeyAcceptedKeyTypes`.

Error de acceso: se denegó la clave pública SSH al conectarse a un repositorio CodeCommit

Problema: cuando intentas utilizar un punto final de SSH para comunicarte con un CodeCommit repositorio, aparece un mensaje de error que contiene la frase. `Error: public key denied`

Posibles soluciones: El motivo más común de este error es que no ha completado la configuración de las conexiones SSH. Configure un par de claves SSH públicas y privadas y, a continuación, asocie la clave pública con su usuario de IAM. Para obtener más información sobre la configuración de SSH, consulte [Para conexiones SSH en Linux, macOS o Unix](#) y [Para conexiones SSH en Windows](#).

Solución de problemas del asistente de credenciales y las conexiones HTTPS a AWS CodeCommit

La siguiente información puede ayudarte a solucionar problemas habituales al utilizar el asistente de credenciales incluido con el HTTPS para conectarte a AWS CLI los repositorios. CodeCommit

Note

Si bien el asistente de credenciales es un método compatible para conectarse CodeCommit mediante un acceso federado, un proveedor de identidad o credenciales temporales, el método recomendado es instalar y usar la utilidad. `git-remote-codecommit` Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Temas

- [Recibo un error al ejecutar el comando `git config` para configurar el ayudante de credenciales](#)
- [Aparece un error al leer el nombre de usuario cuando intento clonar un repositorio](#)
- [Recibo un error de comando no encontrado en Windows al usar el ayudante de credenciales](#)
- [Se me pide un nombre de usuario cuando me conecto a un repositorio CodeCommit](#)
- [Git para macOS: he configurado el ayudante de credenciales correctamente, pero ahora no tengo acceso a mi repositorio \(403\)](#)
- [Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio \(403\)](#)

Recibo un error al ejecutar el comando **git config** para configurar el ayudante de credenciales

Problema: cuando intentas ejecutar el comando `git config` para configurar el asistente de credenciales para que se comuniquen con un CodeCommit repositorio, aparece un error que indica que hay muy pocos argumentos o un indicador de uso que sugiere comandos y sintaxis de `Git config`.

Posibles soluciones: el motivo más común de este error es que se utilizan comillas simples para el comando en un sistema operativo Windows o comillas dobles para el comando en un sistema operativo Linux, macOS o Unix. La sintaxis correcta es la siguiente:

- Windows: `git config --global credential.helper '!aws codecommit credential-helper $@'`
- Linux, macOS o Unix: `git config --global credential.helper '!aws codecommit credential-helper $@'`

Aparece un error al leer el nombre de usuario cuando intento clonar un repositorio

Problema: Al intentar clonar un repositorio con el ayudante de credenciales, aparece un error que indica que el sistema no ha podido leer el nombre de usuario de la URL del repositorio.

Soluciones posibles: El motivo más frecuente de este error es que el archivo `.gitconfig` no se ha creado o configurado correctamente en el ordenador. Abra el archivo `.gitconfig` y asegúrese de que el ayudante de credenciales esté configurado correctamente. Si está en un equipo con un sistema operativo Linux, macOS o Unix, asegúrese también de que el valor de `$HOME` esté configurado correctamente para su sistema.

Recibo un error de comando no encontrado en Windows al usar el ayudante de credenciales

Problema: Tras actualizar la AWS CLI, se produce un error en las conexiones del asistente de credenciales a CodeCommit los repositorios. `aws codecommit credential-helper $@ get: aws: command not found`

Causa: el motivo más común de este error es que la versión de AWS CLI se ha actualizado a una versión que usa Python 3. Hay un problema conocido con el paquete MSI. Para verificar si tiene una de las versiones afectadas, abra una línea de comandos y ejecute el siguiente comando: `aws --version`

Si la versión de Python de salida comienza por 3, tiene una versión afectada. Por ejemplo:

```
aws-cli/1.16.62 Python/3.6.2 Darwin/16.7.0 botocore/1.12.52
```

Soluciones posibles: puede solucionar este tema con una de las siguientes operaciones:

- Instálelo y AWS CLI configúrelo en Windows usando Python y pip en lugar de MSI. Para obtener más información, consulte [Instalar Python, pip y el AWS CLI en Windows](#).

- Edite manualmente el archivo `.gitconfig` para cambiar la sección `[credential]` para que apunte explícitamente a `aws.cmd` en el equipo local. Por ejemplo:

```
[credential]
  helper = !"C:\\Program Files\\Amazon\\AWSCLI\\bin\\aws.cmd\" codecommit
credential-helper $@
UseHttpPath = true
```

- Ejecute el comando `git config` para actualizar el archivo `.gitconfig` con el fin de hacer referencia explícita a `aws.cmd` y actualice manualmente la variable de entorno `PATH` para que incluya la ruta al comando según sea necesario. Por ejemplo:

```
git config --global credential.helper "!aws.cmd codecommit credential-helper $@"
git config --global credential.UseHttpPath true
```

Se me pide un nombre de usuario cuando me conecto a un repositorio CodeCommit

Problema: cuando intentas usar el asistente de credenciales para comunicarte con un CodeCommit repositorio, aparece un mensaje en el que se te pide tu nombre de usuario.

Posibles soluciones: configure su AWS perfil o asegúrese de que el perfil que está utilizando es el que ha configurado para trabajar. CodeCommit Para obtener más información sobre la configuración, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit repositorios en Linux, macOS o Unix con el AWS CLI asistente de credenciales](#) o [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit los repositorios en Windows con el asistente de AWS CLI credenciales](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [Gestión de claves de acceso para usuarios de IAM](#) y [¿Cómo puedo obtener credenciales?](#)

Git para macOS: he configurado el ayudante de credenciales correctamente, pero ahora no tengo acceso a mi repositorio (403)

Problema: en macOS, parece que el ayudante de credenciales no accede o no puede utilizar sus credenciales según lo previsto. Esto puede deberse a dos problemas diferentes:

- AWS CLI Está configurado para una Región de AWS diferente del repositorio en el que existe.
- La utilidad Keychain Access tiene guardadas unas credenciales que ya han caducado.

Posibles soluciones: para comprobar si AWS CLI está configurada para la región correcta, ejecute el `aws configure` comando y revise la información que se muestra. Si el CodeCommit repositorio se encuentra en un repositorio Región de AWS diferente al que se muestra para el archivo AWS CLI, debe ejecutar el `aws configure` comando y cambiar los valores por otros adecuados para esa región. Para obtener más información, consulte [Paso 1: Configuración inicial para CodeCommit](#).

La versión predeterminada de Git publicada en OS X y macOS utiliza la utilidad Keychain Access para guardar las credenciales generadas. Por motivos de seguridad, la contraseña generada para acceder al CodeCommit repositorio es temporal, por lo que las credenciales almacenadas en el llavero dejan de funcionar transcurridos unos 15 minutos. Si solo accedes a Git con CodeCommit, prueba lo siguiente:

1. En Terminal, ejecute el comando `git config` para buscar el archivo de configuración de Git (`gitconfig`) donde está definida la utilidad Keychain Access. En función de su sistema local y preferencias, es posible que tenga más de un archivo `gitconfig`.

```
git config -l --show-origin | grep credential
```

En el resultado de este comando, busque resultados similares a los siguientes:

```
file://path/to/gitconfig credential.helper=osxkeychain
```

El archivo indicado al principio de esta línea es el archivo de configuración de Git que debe editar.

2. Para editar el archivo de configuración de Git, utilice un editor de texto sin formato o ejecute el siguiente comando:

```
nano /usr/local/git/etc/gitconfig
```

3. Modifique la configuración con una de las siguientes estrategias:

- Comente o elimine la sección de credenciales que contiene `helper = osxkeychain`. Por ejemplo:

```
# helper = osxkeychain
```

- Actualice las secciones del ayudante de credenciales `aws credential helper` y `osxkeychain` para que tengan contexto. Por ejemplo, `osxkeychain` se usa para autenticarse GitHub en:

```
[credential "https://git-codecommit.us-east-1.amazonaws.com"]
  helper = !aws --profile CodeCommitProfile codecommit credential-helper $@
  UseHttpPath = true
[credential "https://github.com"]
  helper = osxkeychain
```

En esta configuración, Git usará el ayudante de `osxkeychain` cuando el host remoto coincida con “`https://github.com`” y el ayudante de credenciales cuando el host remoto coincida con “`https://git-codecommit.us-east-1.amazonaws.com`”.

- Incluye un ayudante de cadena vacío antes del ayudante de credenciales. Por ejemplo, para no usar el `osxkeychain` ayudante cuando se usa el perfil nombrado *CodeCommitProfile* con la CLI:

```
[credential]
  helper =
  helper = !aws --profile CodeCommitProfile codecommit credential-helper $@
  UseHttpPath = true
```

Tip

También puede configurar la línea que sigue a la línea auxiliar de cadena vacía para que no coincida CodeCommit si desea que excluya todos los perfiles:

```
helper = !aws codecommit credential-helper $@
```

De forma alternativa, si desea seguir utilizando la utilidad Keychain Access para almacenar en caché credenciales para otros repositorios de Git, modifique el encabezado en lugar de incluir un comentario en la línea. Por ejemplo, para permitir las credenciales en caché GitHub, puede modificar el encabezado de la siguiente manera:

```
[credential "https://github.com"]
  helper = osxkeychain
```


Si accedes a otros repositorios con Git, puedes configurar la utilidad Keychain Access para que no proporcione credenciales para tus CodeCommit repositorios. Para configurar la utilidad Keychain Access:

1. Abra la utilidad Keychain Access. (Puede utilizar la función Finder para localizarlo).
2. Busca `git-codecommit.us-east-2.amazonaws.com` y reemplaza `us-east-2` por el Región de AWS lugar donde se encuentra el repositorio. Marque la fila, abra el menú contextual (haga clic con el botón derecho) y, a continuación, elija Get Info.
3. Elija la pestaña Access Control.
4. En Confirm before allowing access, elija `git-credential-osxkeychain` y, a continuación, seleccione el signo menos para eliminarla de la lista.

 Note

Una vez que haya eliminado `git-credential-osxkeychain` de la lista, puede ver un cuadro de diálogo cada vez que ejecute un comando de Git. Elija Deny para continuar. Si las ventanas emergentes le interrumpen demasiado, aquí tiene algunas opciones alternativas:

- Conéctese para CodeCommit usar credenciales SSH o Git en lugar del asistente de credenciales con HTTPS. Para obtener más información, consulte [Para conexiones SSH en Linux, macOS o Unix](#) y [Configuración de usuarios HTTPS mediante credenciales de Git](#).
- En la utilidad Keychain Access, en la pestaña Access Control de `git-codecommit.us-east-2.amazonaws.com`, elija la opción Allow all applications to access this item (access to this item is not restricted). Esto impide que aparezcan ventanas emergentes, pero las credenciales acabarán caducando (de media, duran unos 15 minutos) y aparece un mensaje de error 403. Cuando suceda esto, elimine el elemento de la cadena de claves para restaurar la funcionalidad.
- Instalar una versión de Git que no utilice la cadena de claves de forma predeterminada, o bien
- Considere una solución de scripting para eliminar el elemento de la cadena de claves. Para ver una muestra de una solución con scripts generada por la comunidad, consulte [Mac OS X Script to Periodically Delete Cached Credentials in the OS X Certificate Store](#) en [Integraciones de productos y servicios](#).

Si desea que Git deje de utilizar la utilidad Keychain Access por completo, puede configurar Git para dejar de utilizar osxkeychain como ayudante de credenciales. Por ejemplo, si abre un terminal y ejecuta el comando `git config --system credential.helper` y devuelve `osxkeychain`, Git está configurado para usar la utilidad Keychain Access. Puede cambiarlo ejecutando el siguiente comando:

```
git config --system --unset credential.helper
```

Ten en cuenta que ejecutar este comando con la `--system` opción cambia el comportamiento de Git en todo el sistema para todos los usuarios, y esto podría tener consecuencias no deseadas para otros usuarios o para otros repositorios si además utilizas otros servicios de repositorio. CodeCommit Tenga en cuenta también que este enfoque podría requerir el uso de `sudo` y que su cuenta podría no tener permisos del sistema suficientes para aplicar este cambio. Asegúrese de verificar que el comando se ha aplicado correctamente ejecutando el comando `git config --system credential.helper` de nuevo. Para obtener más información consulte [Customizing Git - Git Configuration](#) y [este artículo en Stack Overflow](#).

Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio (403)

Problema: En Windows, parece que el ayudante de credenciales no accede o no puede utilizar sus credenciales según lo previsto. Esto puede deberse a diferentes problemas:

- AWS CLI Está configurado para un repositorio Región de AWS diferente de aquel en el que existe el repositorio.
- De forma predeterminada, Git para Windows instala una utilidad Git Credential Manager que no es compatible con CodeCommit las conexiones que utilizan el asistente de AWS credenciales. Cuando se instala, provoca que se produzcan errores en las conexiones al repositorio aunque el ayudante de credenciales se haya instalado con el repositorio y se haya configurado para conectarse a él. AWS CLI CodeCommit
- Es posible que algunas versiones de Git para Windows no cumplan por completo con [RFC 2617](#) y [RFC 4559](#), lo que podría provocar problemas potenciales con las credenciales de Git y el ayudante de credenciales incluido en la AWS CLI. Para obtener más información, consulte [Version 2.11.0\(3\) does not ask for username/password](#).

Posibles soluciones:

- Si estás intentando usar el ayudante de credenciales incluido en el AWS CLI, considera conectarte con las credenciales de Git a través de HTTPS en lugar de usar el ayudante de credenciales. Las credenciales de Git configuradas para su usuario de IAM son compatibles con el administrador de credenciales de Git para Windows, a diferencia del ayudante de credenciales para AWS CodeCommit. Para obtener más información, consulte [Para usuarios HTTPS mediante credenciales de Git](#).

Si quieres usar el asistente de credenciales, para comprobar si AWS CLI está configurado correctamente Región de AWS, ejecuta el `aws configure` comando y revisa la información que se muestra. Si el CodeCommit repositorio se encuentra en un repositorio Región de AWS diferente al que se muestra para el AWS CLI, debe ejecutar el `aws configure` comando y cambiar los valores por otros adecuados para esa región. Para obtener más información, consulte [Paso 1: Configuración inicial para CodeCommit](#).

- Si es posible, desinstale y reinstale Git para Windows. Al instalar Git para Windows, desmarque la casilla de verificación de la opción para instalar la utilidad Administrador de credenciales de Git. Este gestor de credenciales no es compatible con el ayudante de credenciales para AWS CodeCommit. Si instala el Administrador de credenciales de Git u otra utilidad de gestión de credenciales y no desea desinstalarla, puede modificar el archivo `.gitconfig` y añadir la gestión de credenciales específica para CodeCommit:
 1. Abra el Panel de control, elija el Administrador de credenciales y elimine las credenciales almacenadas. CodeCommit
 2. Abra el archivo `.gitconfig` en cualquier editor de texto sin formato, como Notepad.

 Note

Si trabaja con varios perfiles de Git, puede tener tanto archivos `.gitconfig` locales como globales. Asegúrese de editar el archivo correspondiente.

3. Añada la siguiente sección a su archivo `.gitconfig`:

```
[credential "https://git-codecommit.*.amazonaws.com"]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

4. Guarde el archivo y, a continuación, abra una nueva sesión de línea de comandos antes de intentar conectarse de nuevo.

También puede utilizar este enfoque si desea utilizar el asistente de credenciales AWS CodeCommit cuando se conecte a los CodeCommit repositorios y otro sistema de administración de credenciales cuando se conecte a otros repositorios alojados, como los repositorios. GitHub

Para restablecer el ayudante de credenciales que se usa de forma predeterminada, puede utilizar la opción `--system` en lugar de `--global` o `--local` al ejecutar el comando `git config`.

- Si utiliza credenciales de Git en un equipo Windows, puede intentar solucionar los problemas de incumplimiento RFC incluyendo su nombre de usuario de Git en la cadena de conexión. Por ejemplo, para solucionar el problema y clonar un repositorio cuyo nombre *MyDemoRepo* pertenezca a la región EE.UU. Este (Ohio):

```
git clone https://Your-Git-Credential-Username@git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

Note

Este enfoque no funciona si tiene un carácter `@` en el nombre de usuario de sus credenciales Git. Deberá codificar la dirección URL (también conocida como escape URL o [código por ciento](#)) del carácter.

Solución de problemas de clientes Git y AWS CodeCommit

La siguiente información puede ayudarle a solucionar los problemas más frecuentes al utilizar Git con los repositorios de AWS CodeCommit. Para solucionar los problemas relacionado con los clientes de Git cuando utilizan HTTPS o SSH, consulte también [Solución de problemas de credenciales de Git \(HTTPS\)](#), [Solución de problemas de conexiones SSH](#) y [Solución de problemas del ayudante de credenciales \(HTTPS\)](#).

Temas

- [Error de Git: error: error de RPC; resultado = 56, código HTTP = 200 grave: El extremo remoto se ha interrumpido de forma inesperada](#)
- [Error de Git: demasiados comandos de actualización de referencia](#)
- [Error de Git: envío a través de HTTPS dividido en algunas versiones de Git](#)
- [Error de Git: 'gnutls_handshake\(\) failed'](#)

- [Error de Git: Git no encuentra el CodeCommit repositorio o no tiene permiso para acceder al repositorio](#)
- [Git en Windows: métodos de autenticación no admitidos disponibles \(clave pública\)](#)

Error de Git: error: error de RPC; resultado = 56, código HTTP = 200 grave: El extremo remoto se ha interrumpido de forma inesperada

Problema: al enviar un cambio grande, un gran número de cambios o un repositorio grande, a menudo las conexiones HTTPS de ejecución prolongada suelen interrumpirse de forma prematura debido a problemas de red o de configuración del cortafuegos.

Soluciones posibles: Envíe con SSH o al migrar un repositorio grande, siga los pasos que se detallan en [Migrar de forma incremental un repositorio](#). Asimismo, asegúrese de no superar los límites de tamaño de los archivos individuales. Para obtener más información, consulte [Cuotas](#).

Error de Git: demasiados comandos de actualización de referencia

Problema: el número máximo de actualizaciones de referencia por envío es 4 000. Este error aparece cuando el envío contiene más de 4 000 actualizaciones de referencia.

Soluciones posibles: intente enviar ramificaciones y etiquetas de forma individual con `git push --all` y `git push --tags`. Si tiene demasiadas etiquetas, divídalas en múltiples envíos. Para obtener más información, consulte [Cuotas](#).

Error de Git: envío a través de HTTPS dividido en algunas versiones de Git

Problema: ha surgido un problema con la actualización de curl a 7.41.0 lo que provoca que la autenticación Digest basada en SSPI devuelva un error. Las versiones de Git afectadas son 1.9.5.msysgit.1. También es posible que algunas versiones de Git para Windows no cumplan por completo las normas [RFC 2617](#) y [RFC 4559](#), lo que podría provocar problemas potenciales con las conexiones HTTPS mediante credenciales de Git o la aplicación ayudante de credenciales incluido en la AWS CLI.

Soluciones posibles: compruebe los problemas conocidos de su versión de Git o utilice una versión anterior o posterior. Para obtener más información sobre msysgit, consulta [Push to HTTPS Is Broken](#) en los foros. GitHub Para obtener más información acerca de los problemas de las versiones de Git para Windows, consulte [Version 2.11.0\(3\) does not ask for username/password](#).

Error de Git: 'gnutls_handshake() failed'

Problema: en Linux, cuando intentas usar Git para comunicarte con un CodeCommit repositorio, aparece un mensaje de error que contiene la frase `error: gnutls_handshake() failed`.

Soluciones posibles: recopile Git en relación con OpenSSL. Para un enfoque, consulte ["Error: gnutls_handshake\(\) failed" When Connecting to HTTPS Servers](#) en los foros de Ask Ubuntu.

Como alternativa, puedes usar SSH en lugar de HTTPS para comunicarte con los CodeCommit repositorios.

Error de Git: Git no encuentra el CodeCommit repositorio o no tiene permiso para acceder al repositorio

Problema: una barra al final de la cadena de conexión puede provocar que los intentos de conexión devuelvan un error.

Soluciones posibles: asegúrese de que ha proporcionado el nombre y la cadena de conexión correctos para el repositorio y de que no haya ninguna barra al final. Para obtener más información, consulte [Conectarse a un repositorio](#).

Git en Windows: métodos de autenticación no admitidos disponibles (clave pública)

Problema: después de configurar el acceso de SSH para Windows, aparece un error de acceso denegado al intentar usar comandos como `git pull`, `git push` o `git clone`.

Soluciones posibles: la causa más común de este error es que su equipo tenga una variable de entorno `GIT_SSH` y que se haya configurado para admitir otra utilidad de conexión, como PuTTY. Para solucionar este problema, pruebe uno de los siguientes pasos:

- Abra un emulador de Bash y añada el parámetro `GIT_SSH_COMMAND="ssh"` delante del comando de Git. Por ejemplo, si intenta clonar a un repositorio, en lugar de ejecutar `git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo`, ejecute:

```
GIT_SSH_COMMAND="ssh" git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/
repos/MyDemoRepo my-demo-repo
```

- Renombre o elimine la variable de entorno `GIT_SSH` si ya no la utiliza. A continuación, abra una nueva sesión del símbolo del sistema o del emulador de Bash y pruebe el comando de nuevo.

Para obtener más información acerca de la solución de problemas relacionados con Git en Windows cuando se utiliza SSH, consulte [Solución de problemas de conexiones SSH](#).

Solución de problemas de acceso y AWS CodeCommit

La siguiente información puede ayudarle a solucionar los errores de acceso al conectarse a los repositorios de AWS CodeCommit .

Temas

- [Error de acceso: se me pide un nombre de usuario y una contraseña cuando me conecto a un CodeCommit repositorio desde Windows](#)
- [Error de acceso: clave pública denegada al conectarse a un repositorio CodeCommit](#)
- [Error de acceso: aparece el mensaje «Se ha superado la velocidad» o «429» al conectarse a un repositorio CodeCommit](#)

Error de acceso: se me pide un nombre de usuario y una contraseña cuando me conecto a un CodeCommit repositorio desde Windows

Problema: cuando intentas usar Git para comunicarte con un CodeCommit repositorio, aparece un cuadro de diálogo que te pide tu nombre de usuario y contraseña.

Soluciones posibles: podría tratarse del sistema de gestión de credenciales integrado de Windows. Haga lo siguiente en función de la configuración:

- Si utiliza HTTPS con credenciales de Git, sus credenciales de Git aún no están almacenadas en el sistema. Proporcione las credenciales de Git y continúe. No se le volverá a preguntar. Para obtener más información, consulte [Para usuarios HTTPS mediante credenciales de Git](#).
- Si utilizas HTTPS con el asistente de credenciales AWS CodeCommit, no es compatible con el sistema de administración de credenciales de Windows. Elija Cancelar.

También podría indicar que ha instalado el Administrador de credenciales de Git al instalar Git para Windows. El administrador de credenciales de Git no es compatible con el ayudante de credenciales que CodeCommit se incluye en. AWS CLI Considere desinstalar el Administrador de credenciales de Git. También puede instalar y configurar git-remote-codecommit como alternativa al uso de la aplicación auxiliar de credenciales de CodeCommit.

Para obtener más información, consulte [Pasos de configuración para las conexiones HTTPS a AWS CodeCommit con git-remote-codecommit](#), [Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI](#) y [Git para Windows: he instalado Git para Windows, pero no tengo permiso para acceder a mi repositorio \(403\)](#).

Error de acceso: clave pública denegada al conectarse a un repositorio CodeCommit

Problema: cuando intentas utilizar un punto final SSH para comunicarte con un CodeCommit repositorio, aparece un mensaje de error que contiene la frase `Error: public key denied`.

Posibles soluciones: El motivo más común de este error es que no ha completado la configuración de las conexiones SSH. Configure un par de claves SSH públicas y privadas y, a continuación, asocie la clave pública con su usuario de IAM. Para obtener más información sobre la configuración de SSH, consulte [Para conexiones SSH en Linux, macOS o Unix](#) y [Para conexiones SSH en Windows](#).

Error de acceso: aparece el mensaje «Se ha superado la velocidad» o «429» al conectarse a un repositorio CodeCommit

Problema: cuando intenta comunicarse con un repositorio de CodeCommit, aparece un mensaje que indica "Tasa superada" o un código de error "429". La comunicación se ralentiza significativamente o falla.

Causa: todas las llamadas CodeCommit, ya sean desde una aplicación AWS CLI, el cliente Git o el Consola de administración de AWS, están sujetas a un número máximo de solicitudes por segundo y a un total de solicitudes activas. No puedes superar la tasa máxima de solicitudes permitida para una cuenta de Amazon Web Services en ninguna cuenta Región de AWS. Si las solicitudes superan la tasa máxima, recibe un error y las llamadas adicionales se limitan temporalmente para su cuenta de Amazon Web Services. Durante el periodo de limitación controlada, las conexiones a CodeCommit se ralentizan y podrían no realizarse.

Posibles soluciones: tome medidas para reducir el número de conexiones o llamadas CodeCommit o para dispersar las solicitudes. Algunos enfoques que podría tener en cuenta:

- Implementar la fluctuación en las solicitudes, especialmente en las solicitudes de sondeo periódicas

Si tiene una aplicación que sondea CodeCommit periódicamente y esta aplicación se ejecuta en varias instancias de Amazon EC2, introduzca la fluctuación (una cantidad de retraso aleatoria) para que las distintas instancias de Amazon EC2 no sondeen en el mismo segundo. Recomendamos un número aleatorio de 0 a 59 segundos para distribuir uniformemente los mecanismos de sondeo a lo largo de un minuto.

- Usar una arquitectura basada en eventos en lugar de sondeo

En lugar de sondear, utilice una arquitectura basada en eventos para que las llamadas solo se realicen cuando ocurra un evento. Considere la posibilidad de utilizar las notificaciones de CloudWatch [AWS CodeCommit eventos para activar su flujo](#) de trabajo.

- Implemente reintentos de error y retrasos exponenciales para acciones de Git automatizadas APIs

Los reintentos de errores y los retardos exponenciales pueden ser de ayuda para limitar la tasa de llamadas. Cada AWS SDK implementa la lógica de reintento automático y los algoritmos de retroceso exponencial. Para realizar los comandos git push y git pull, es posible que deba implementar su propia lógica de reintentos. Para obtener más información, consulte [Reintentos de error](#) y retrocesos exponenciales en. AWS

- Solicita un aumento de la cuota de CodeCommit servicio en el AWS Support Center

Para obtener un aumento del límite de servicio, debe confirmar que ya ha seguido las sugerencias que se ofrecen aquí, incluida la implementación de reintentos de error o métodos de retardo exponencial. En su solicitud, también debe proporcionar la Región de AWS cuenta de Amazon Web Services y el plazo afectados por los problemas de limitación.

Solución de problemas de configuración y AWS CodeCommit

La siguiente información puede ayudarle a solucionar los errores de configuración que pueden aparecer al conectarse a los AWS CodeCommit repositorios.

Temas

- [Error de configuración: no se pueden configurar AWS CLI las credenciales en macOS](#)

Error de configuración: no se pueden configurar AWS CLI las credenciales en macOS

Problema: cuando se ejecuta `aws configure` para configurar el AWS CLI, aparece un `ConfigParseError` mensaje.

Soluciones posibles: la causa más común de este error es que ya exista un archivo de credenciales. Navegue hasta `~/.aws` y busque un archivo denominado `credentials`. Cambie el nombre o elimine dicho archivo y, a continuación, ejecute `aws configure` de nuevo.

Solución de problemas de la consola y AWS CodeCommit

La siguiente información puede ayudarle a solucionar los errores de la consola al utilizar los repositorios de AWS CodeCommit .

Temas

- [Error de acceso: se denegó el acceso a la clave de cifrado de un CodeCommit repositorio desde la consola o AWS CLI](#)
- [Error de cifrado: no se puede descifrar el repositorio](#)
- [Error de consola: no se puede examinar el código de un CodeCommit repositorio desde la consola](#)
- [Error de visualización: no se puede ver un archivo o una comparación entre archivos](#)

Error de acceso: se denegó el acceso a la clave de cifrado de un CodeCommit repositorio desde la consola o AWS CLI

Problema: Al intentar acceder CodeCommit desde la consola o desde la AWS CLI, aparece un mensaje de error que contiene la frase `EncryptionKeyAccessDeniedException` o `User is not authorized for the KMS default key for CodeCommit 'aws/codecommit' in your account`.

Posibles soluciones: la causa más común de este error es que su cuenta de Amazon Web Services no está suscrita AWS Key Management Service, lo cual es obligatorio para CodeCommit. Abra la AWS KMS consola, selecciona claves AWS administradas y, a continuación, selecciona Comenzar ahora. Si aparece un mensaje en el que se indica que actualmente no estás suscrito al AWS Key Management Service servicio, sigue las instrucciones de esa página para suscribirte. Para obtener más información sobre CodeCommit y AWS Key Management Service, consulte [AWS KMS y cifrado](#).

Error de cifrado: no se puede descifrar el repositorio

Problema: Al intentar acceder a un CodeCommit repositorio desde la consola o el AWS CLI, aparece un mensaje de error que contiene la frase `Repository can't be decrypted`.

Posibles soluciones: la causa más común de este error es que la AWS KMS clave utilizada para cifrar y descifrar los datos de este repositorio no está activa o está pendiente de ser eliminada. Para ello, AWS Key Management Service es necesaria una clave activa Clave administrada de AWS o gestionada por el cliente. CodeCommit abra la AWS KMS consola, elija entre Claves administradas por AWS claves administradas por el cliente y asegúrese de que la clave utilizada para el repositorio esté presente en el Región de AWS lugar donde se encuentra el repositorio y de que su estado sea Activo. Para obtener más información sobre CodeCommit y AWS Key Management Service, consulte [AWS KMS y cifrado](#).

Important

Si la clave que se utilizó para cifrar y descifrar los datos del repositorio se ha eliminado permanentemente o no se puede acceder a ella por algún otro motivo, no se podrá acceder a los datos de los repositorios cifrados con esa clave.

Error de consola: no se puede examinar el código de un CodeCommit repositorio desde la consola

Problema: al intentar examinar el contenido de un repositorio desde la consola, aparecerá un mensaje de error denegando el acceso.

Posibles soluciones: la causa más común de este error es que una política de IAM aplicada a tu cuenta de Amazon Web Services deniega uno o varios de los permisos necesarios para navegar por el código desde la CodeCommit consola. Para obtener más información sobre los permisos de CodeCommit acceso y la navegación, consulte [Autenticación y control de acceso para AWS CodeCommit](#).

Error de visualización: no se puede ver un archivo o una comparación entre archivos

Problema: al intentar ver un archivo o comparar dos versiones de un archivo en la CodeCommit consola, aparece un error que indica que el archivo o la diferencia son demasiado grandes para mostrarlos.

Soluciones posibles: la causa más común de este error es que el archivo es demasiado grande para mostrarse, contiene una o más líneas que superan el límite de caracteres de una sola línea del archivo o que la diferencia entre las dos versiones del archivo supera el límite de líneas. Para obtener más información, consulte [Cuotas](#). Para ver el archivo o las diferencias entre las versiones del archivo, puede abrir el archivo localmente en el IDE que prefiera, utilizar una herramienta de diferencias de Git o ejecutar el comando `git diff`.

Solución de problemas desencadenantes y AWS CodeCommit

La siguiente información puede ayudarle a solucionar problemas con disparadores en AWS CodeCommit.

Temas

- [Error de disparador: un disparador del repositorio no se ejecuta cuando se espera](#)

Error de disparador: un disparador del repositorio no se ejecuta cuando se espera

Problema: al parecer uno o más disparadores configurados en un repositorio no se ejecutan o no se ejecutan como se espera.

Soluciones posibles: en caso de que el objetivo del disparador sea una función AWS Lambda , asegúrese de que haya configurado la política de recursos de la función para obtener acceso a CodeCommit.

De forma alternativa, edite el disparador y asegúrese de que los eventos para los que desea disparar acciones estén seleccionados y que las ramificaciones del disparador incluyan la ramificación donde quiere ver las respuestas a las acciones. Intente cambiar la configuración del disparador por All repository events y All branches y, a continuación, pruebe el disparador. Para obtener más información, consulte [Edición de los disparadores de un repositorio](#).

Activar la depuración

Problema: quiero activar la depuración para obtener más información sobre el repositorio y sobre cómo ejecuta Git los comandos.

Soluciones posibles: pruebe lo siguiente:

1. En el terminal o en el símbolo del sistema, ejecute los siguientes comandos en el equipo local antes de ejecutar comandos Git:

En Linux, macOS o Unix:

```
export GIT_TRACE_PACKET=1
export GIT_TRACE=1
export GIT_CURL_VERBOSE=1
```

En Windows:

```
set GIT_TRACE_PACKET=1
set GIT_TRACE=1
set GIT_CURL_VERBOSE=1
```

Note

Establecer `GIT_CURL_VERBOSE` solo es útil para las conexiones HTTPS. SSH no utiliza la biblioteca `libcurl`.

2. Para obtener más información sobre su repositorio de Git, le recomendamos instalar la última versión de [git-sizer](#). Siga las instrucciones para instalar la utilidad adecuada para su sistema operativo y entorno. Una vez instalada, en la línea de comandos o en el terminal, cambie los directorios al repositorio local y, a continuación, ejecute el siguiente comando:

```
git-sizer --verbose
```

 Tip

Plantéese guardar el resultado del comando en un archivo para poder compartirlo fácilmente con otras personas a la hora de solucionar problemas, especialmente en el futuro.

AWS CodeCommit Referencia de

Los siguientes temas de referencia pueden ayudarte a entender mejor Git CodeCommit Regiones de AWS, los límites de servicio y mucho más.

Temas

- [Regiones y puntos de conexión de Git para AWS CodeCommit](#)
- [Utilización AWS CodeCommit con puntos finales de VPC de interfaz](#)
- [Cuotas en AWS CodeCommit](#)
- [AWS CodeCommit referencia de línea de comandos](#)
- [Comandos básicos de Git](#)

Regiones y puntos de conexión de Git para AWS CodeCommit

Cada CodeCommit repositorio está asociado a un. Región de AWS CodeCommit ofrece puntos finales regionales para realizar sus solicitudes al servicio. Además, CodeCommit proporciona puntos de conexión Git para los protocolos SSH y HTTPS en todas las regiones donde CodeCommit esté disponible.

Todos los ejemplos de esta guía utilizan la misma URL del punto de conexión para Git en este de EE. UU. (Ohio): `git-codecommit.us-east-2.amazonaws.com`. Sin embargo, cuando utilices Git y configures tus conexiones, asegúrate de elegir el punto de conexión de Git que coincida con el Región de AWS que aloja tu CodeCommit repositorio. Por ejemplo, si desea conectarse a un repositorio en este de EE. UU. (Norte de Virginia), utilice la URL del punto de conexión de `git-codecommit.us-east-1.amazonaws.com`. Lo mismo sucede con las llamadas a la API. Cuando hagas conexiones a un CodeCommit repositorio con el AWS CLI o los SDK, asegúrate de usar el punto de conexión regional correcto para el repositorio.

Temas

- [compatible Regiones de AWS para CodeCommit](#)
- [Puntos de conexión de Git](#)
- [Huellas digitales del servidor para CodeCommit](#)

compatible Regiones de AWS para CodeCommit

Puede crear y usar CodeCommit repositorios de las siguientes maneras: Regiones de AWS

- Este de EE. UU. (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- EE.UU. Oeste (Oregón)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Europa (Fráncfort)
- Europa (Estocolmo)
- Europa (Milán)
- África (Ciudad del Cabo)
- Israel (Tel Aviv)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Yakarta)
- Medio Oriente (EAU)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Mumbai)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Hong Kong)
- América del Sur (São Paulo)
- Middle East (Bahrain)
- Canadá (centro)
- China (Pekín)
- China (Ningxia)
- AWS GovCloud (US-West)

- AWS GovCloud (US-East)

CodeCommit ha añadido compatibilidad con la norma gubernamental Federal de Procesamiento de Información (FIPS), publicación 140-2, en algunas regiones. Para obtener más información acerca de los puntos de enlace de FIPS y FIPS, consulte la [descripción general del Estándar de Procesamiento de la Información Federal \(FIPS\) 140-2](#). Para ver los puntos de enlace de conexión de Git que admiten FIPS, consulte [Puntos de conexión de Git](#).

Para obtener más información sobre los puntos de conexión regionales AWS CLI, los servicios y las llamadas a las API CodeCommit, consulte Puntos de conexión y cuotas [AWS CodeCommit](#).

Puntos de conexión de Git

Utilice las siguientes URL para configurar las conexiones de Git a los repositorios de CodeCommit:

Puntos finales de conexión de Git para AWS CodeCommit

Nombre de la región	Region	URL del punto de conexión	Protocolo
Este de EE. UU. (Ohio)	us-east-2	https://git-codecommit.us-east-2.amazonaws.com	HTTPS
Este de EE. UU. (Ohio)	us-east-2	ssh://git-codecommit.us-east-2.amazonaws.com	SSH
Este de EE. UU. (Ohio)	us-east-2	https://git-codecommit-fips.us-east-2.amazonaws.com	HTTPS
Este de EE. UU. (Norte de Virginia)	us-east-1	https://git-codecommit.us-east-1.amazonaws.com	HTTPS
Este de EE. UU. (Norte de Virginia)	us-east-1	ssh://git-codecommit.us-east-1.amazonaws.com	SSH

Nombre de la región	Region	URL del punto de conexión	Protocolo
Este de EE. UU. (Norte de Virginia)	us-east-1	<code>https://git-codecommit-fips.us-east-1.amazonaws.com</code>	HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	<code>https://git-codecommit.us-west-2.amazonaws.com</code>	HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	<code>ssh://git-codecommit.us-west-2.amazonaws.com</code>	SSH
Oeste de EE. UU. (Oregón)	us-west-2	<code>https://git-codecommit-fips.us-west-2.amazonaws.com</code>	HTTPS
Oeste de EE. UU. (Norte de California)	us-west-1	<code>https://git-codecommit.us-west-1.amazonaws.com</code>	HTTPS
Oeste de EE. UU. (Norte de California)	us-west-1	<code>ssh://git-codecommit.us-west-1.amazonaws.com</code>	SSH
Oeste de EE. UU. (Norte de California)	us-west-1	<code>https://git-codecommit-fips.us-west-1.amazonaws.com</code>	HTTPS
Europa (Irlanda)	eu-west-1	<code>https://git-codecommit.eu-west-1.amazonaws.com</code>	HTTPS
Europa (Irlanda)	eu-west-1	<code>ssh://git-codecommit.eu-west-1.amazonaws.com</code>	SSH

Nombre de la región	Region	URL del punto de conexión	Protocolo
Asia-Pacífico (Tokio)	ap-northeast-1	https://git-codecommit.ap-northeast-1.amazonaws.com	HTTPS
Asia-Pacífico (Tokio)	ap-northeast-1	ssh://git-codecommit.ap-northeast-1.amazonaws.com	SSH
Asia-Pacífico (Singapur)	ap-southeast-1	https://git-codecommit.ap-southeast-1.amazonaws.com	HTTPS
Asia-Pacífico (Singapur)	ap-southeast-1	ssh://git-codecommit.ap-southeast-1.amazonaws.com	SSH
Asia-Pacífico (Sídney)	ap-southeast-2	https://git-codecommit.ap-southeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	ssh://git-codecommit.ap-southeast-2.amazonaws.com	SSH
Asia-Pacífico (Yakarta)	ap-southeast-3	https://git-codecommit.ap-southeast-3.amazonaws.com	HTTPS
Asia-Pacífico (Yakarta)	ap-southeast-3	ssh://git-codecommit.ap-southeast-3.amazonaws.com	SSH
Medio Oriente (EAU)	me-central-1	https://git-codecommit.me-central-1.amazonaws.com	HTTPS

Nombre de la región	Region	URL del punto de conexión	Protocolo
Medio Oriente (EAU)	me-central-1	ssh://git-codecommit.me-central-1.amazonaws.com	SSH
Europa (Fráncfort)	eu-central-1	https://git-codecommit.eu-central-1.amazonaws.com	HTTPS
Europa (Fráncfort)	eu-central-1	ssh://git-codecommit.eu-central-1.amazonaws.com	SSH
Asia-Pacífico (Seúl)	ap-northeast-2	https://git-codecommit.ap-northeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Seúl)	ap-northeast-2	ssh://git-codecommit.ap-northeast-2.amazonaws.com	SSH
América del Sur (São Paulo)	sa-east-1	https://git-codecommit.sa-east-1.amazonaws.com	HTTPS
América del Sur (São Paulo)	sa-east-1	ssh://git-codecommit.sa-east-1.amazonaws.com	SSH
Europa (Londres)	eu-west-2	https://git-codecommit.eu-west-2.amazonaws.com	HTTPS
Europa (Londres)	eu-west-2	ssh://git-codecommit.eu-west-2.amazonaws.com	SSH

Nombre de la región	Region	URL del punto de conexión	Protocolo
Asia-Pacífico (Mumbai)	ap-south-1	https://git-codecommit.ap-south-1.amazonaws.com	HTTPS
Asia-Pacífico (Mumbai)	ap-south-1	ssh://git-codecommit.ap-south-1.amazonaws.com	SSH
Asia-Pacífico (Hyderabad)	ap-south-2	https://git-codecommit.ap-south-2.amazonaws.com	HTTPS
Asia-Pacífico (Hyderabad)	ap-south-2	ssh://git-codecommit.ap-south-2.amazonaws.com	SSH
Canadá (centro)	ca-central-1	https://git-codecommit.ca-central-1.amazonaws.com	HTTPS
Canadá (centro)	ca-central-1	ssh://git-codecommit.ca-central-1.amazonaws.com	SSH
Canadá (centro)	ca-central-1	https://git-codecommit-fips.ca-central-1.amazonaws.com	HTTPS
Europa (París)	eu-west-3	https://git-codecommit.eu-west-3.amazonaws.com	HTTPS
Europa (París)	eu-west-3	ssh://git-codecommit.eu-west-3.amazonaws.com	SSH

Nombre de la región	Region	URL del punto de conexión	Protocolo
AWS GovCloud (US-West)	us-gov-west-1	https://git-codecommit.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (US-West)	us-gov-west-1	ssh://git-codecommit.us-gov-west-1.amazonaws.com	SSH
AWS GovCloud (US-West)	us-gov-west-1	https://git-codecommit-fips.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (US-East)	us-gov-east-1	https://git-codecommit.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-East)	us-gov-east-1	ssh://git-codecommit.us-gov-east-1.amazonaws.com	SSH
AWS GovCloud (US-East)	us-gov-east-1	https://git-codecommit-fips.us-gov-east-1.amazonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	https://git-codecommit.eu-north-1.amazonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	ssh://git-codecommit.eu-north-1.amazonaws.com	SSH
Medio Oriente (Baréin)	me-south-1	https://git-codecommit.me-south-1.amazonaws.com	HTTPS

Nombre de la región	Region	URL del punto de conexión	Protocolo
Medio Oriente (Baréin)	me-south-1	ssh://git-codecommit.me-south-1.amazonaws.com	SSH
Asia-Pacífico (Hong Kong)	ap-east-1	https://git-codecommit.ap-east-1.amazonaws.com	HTTPS
Asia-Pacífico (Hong Kong)	ap-east-1	ssh://git-codecommit.ap-east-1.amazonaws.com	SSH
China (Pekín)	cn-north-1	https://git-codecommit.cn-north-1.amazonaws.com.cn	HTTPS
China (Pekín)	cn-north-1	ssh://git-codecommit.cn-north-1.amazonaws.com.cn	SSH
China (Ningxia)	cn-northwest-1	https://git-codecommit.cn-northwest-1.amazonaws.com.cn	HTTPS
China (Ningxia)	cn-northwest-1	ssh://git-codecommit.cn-northwest-1.amazonaws.com.cn	SSH
Europa (Milán)	eu-south-1	https://git-codecommit.eu-south-1.amazonaws.com	HTTPS
Europa (Milán)	eu-south-1	ssh://git-codecommit.eu-south-1.amazonaws.com	SSH

Nombre de la región	Region	URL del punto de conexión	Protocolo
Asia-Pacífico (Osaka)	ap-northeast-3	https://git-codecommit.ap-northeast-3.amazonaws.com	HTTPS
Asia-Pacífico (Osaka)	ap-northeast-3	ssh://git-codecommit.ap-northeast-3.amazonaws.com	SSH
África (Ciudad del Cabo)	af-south-1	https://git-codecommit.af-south-1.amazonaws.com	HTTPS
África (Ciudad del Cabo)	af-south-1	ssh://git-codecommit.af-south-1.amazonaws.com	SSH
Israel (Tel Aviv)	il-central-1	https://git-codecommit.il-central-1.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	ssh://git-codecommit.il-central-1.amazonaws.com	SSH

Huellas digitales del servidor para CodeCommit

En la siguiente tabla se muestran las huellas digitales públicas para los puntos de enlace de conexión de Git en CodeCommit. Estas huellas digitales del servidor se muestran como parte del proceso de verificación para añadir un punto de conexión a su archivo de hosts conocidos.

Huellas digitales públicas para CodeCommit

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.us-east-2.amazonaws.com	MD5	a9:6d:03:ed:08:42: 21:be:06:e1:e0:2a: d1:75:31:5e
git-codecommit.us-east-2.amazonaws.com	SHA256	31B1W2g5xn/NA2Ck6d yeJIrQ0Wvn7n8UEs56 fG6ZIZQ
git-codecommit.us-east-1.amazonaws.com	MD5	a6:9c:7d:bc:35:f5: d4:5f:8b:ba:6f:c8: bc:d4:83:84
git-codecommit.us-east-1.amazonaws.com	SHA256	eLMY1j0DKA4uvDZc1/ KgtIayZANwX6t8+8is PtotBoY
git-codecommit.us-west-2.amazonaws.com	MD5	a8:68:53:e3:99:ac: 6e:d7:04:7e:f7:92: 95:77:a9:77
git-codecommit.us-west-2.amazonaws.com	SHA256	0pJx9SQpkbPUAHwy58 UVIq0IHcyo1fwCp00u VgcAWPo
git-codecommit.eu-west-1.amazonaws.com	MD5	93:42:36:ea:22:1f: f1:0f:20:02:4a:79: ff:ea:12:1d
git-codecommit.eu-west-1.amazonaws.com	SHA256	tKjRk0L8dmJyTmSbeS dN1S8F/f0iq13Rlvqg TOP1UyQ
git-codecommit.ap-northeast-1.amazonaws.com	MD5	8e:a3:f0:80:98:48: 1c:5c:6f:59:db:a7: 8f:6e:c6:cb

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.ap-northeast-1.amazonaws.com	SHA256	Xk/WeYD/K/bnBybzhi uu4dWpBJtXPf7E30jH U7se40w
git-codecommit.ap-southeast-1.amazonaws.com	MD5	65:e5:27:c3:09:68: 0d:8e:b7:6d:94:25: 80:3e:93:cf
git-codecommit.ap-southeast-1.amazonaws.com	SHA256	ZIsVa70VzxrtIf+Rk4 UbhPv6Es22mSB3uTBo jfPXIno
git-codecommit.ap-southeast-2.amazonaws.com	MD5	7b:d2:c1:24:e6:91: a5:7b:fa:c1:0c:35: 95:87:da:a0
git-codecommit.ap-southeast-2.amazonaws.com	SHA256	nYp+gHas80HY3DqbP4 yanCDFhqDVjseefVbH EXqH2Ec
git-codecommit.ap-southeast-3.amazonaws.com	MD5	64:d9:e0:53:19:4f: a8:91:9a:c3:53:22: a6:a8:ed:a6
git-codecommit.ap-southeast-3.amazonaws.com	SHA256	ATdkGSFhpqIu7RqUVT /1RZo6MLxxxUW9NoDV MbAc/6g
git-codecommit.me-central-1.amazonaws.com	MD5	bd:fa:e2:f9:05:84: d6:39:6f:bc:d6:8d: fe:de:61:76
git-codecommit.me-central-1.amazonaws.com	SHA256	grceUDWubo4MzG1Noa KZKUfrgPvfN3ijli0n Qr11TZA

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.eu-central-1.amazonaws.com	MD5	74:5a:e8:02:fc:b2: 9c:06:10:b4:78:84: 65:94:22:2d
git-codecommit.eu-central-1.amazonaws.com	SHA256	MwGrkiEki8QkkBt1Ag XbYt0hoZYBnZF62VY5 RzGJEUy
git-codecommit.ap-northeast-2.amazonaws.com	MD5	9f:68:48:9b:5f:fc: 96:69:39:45:58:87: 95:b3:69:ed
git-codecommit.ap-northeast-2.amazonaws.com	SHA256	eegAPQrWY9YsYo9ZHI K0mxetfXBHzAZd8Eya 53Qcwko
git-codecommit.sa-east-1.amazonaws.com	MD5	74:99:9d:ff:2b:ef: 63:c6:4b:b4:6a:7f: 62:c5:4b:51
git-codecommit.sa-east-1.amazonaws.com	SHA256	kW+VKB0jpRaG/ZbXkg btMQbKgEDK7JnISV3S VoyCmzU
git-codecommit.us-west-1.amazonaws.com	MD5	3b:76:18:83:13:2c: f8:eb:e9:a3:d0:51: 10:32:e7:d1
git-codecommit.us-west-1.amazonaws.com	SHA256	gzauWTWDXDK2u5KuMMi 5vbKTmfyerdIwgSbzY B0DLpzg
git-codecommit.eu-west-2.amazonaws.com	MD5	a5:65:a6:b1:84:02: b1:95:43:f9:0e:de: dd:ed:61:d3

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.eu-west-2.amazonaws.com	SHA256	r0Rwz5k/IHp/QyrRnf iM9j02D5UEqMbtFNTu DG2hNbs
git-codecommit.ap-south-1.amazonaws.com	MD5	da:41:1e:07:3b:9e: 76:a0:c5:1e:64:88: 03:69:86:21
git-codecommit.ap-south-1.amazonaws.com	SHA256	hUKwnTj7+Xpx4Kddb6 p45j4RazIJ4IhAMD8k 29it0fE
git-codecommit.ap-south-2.amazonaws.com	MD5	bc:cc:9f:15:f8:f3: 58:a2:68:65:21:e2: 23:71:8d:ce
git-codecommit.ap-south-2.amazonaws.com	SHA256	Xe0CyZE0vgR5Xa2YUG qf+jn8/Ut7l7nX/Cms lSFNEig
git-codecommit.ca-central-1.amazonaws.com	MD5	9f:7c:a2:2f:8c:b5: 74:fd:ab:b7:e1:fd: af:46:ed:23
git-codecommit.ca-central-1.amazonaws.com	SHA256	Qz5puafQdANVprLlj6 r0Qyh4lCNsF6ob61dG cPtFS7w
git-codecommit.eu-west-3.amazonaws.com	MD5	1b:7f:97:dd:d7:76: 8a:32:2c:bd:2c:7b: 33:74:6a:76
git-codecommit.eu-west-3.amazonaws.com	SHA256	uw7c2FL564jVoFgtc+ ikzILnKBsZz7t9+CFd SJjKbLI

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.us-gov-west-1.amazonaws.com	MD5	9f:6c:19:3b:88:cd: e8:88:1b:9c:98:6a: 95:31:8a:69
git-codecommit.us-gov-west-1.amazonaws.com	SHA256	djXQoSIFcg8vHe0KVH 1xW/g0F9X37tWTqu4H kng75x4
git-codecommit.us-gov-east-1.amazonaws.com	MD5	00:8d:b5:55:6f:05: 78:05:ed:ea:cb:3f: e6:f0:62:f2
git-codecommit.us-gov-east-1.amazonaws.com	SHA256	fVb+R0z7qW7minenW+ rUpAABRCRBTCzmETAJ EQrg98
git-codecommit.eu-north-1.amazonaws.com	MD5	8e:53:d8:59:35:88: 82:fd:73:4b:60:8a: 50:70:38:f4
git-codecommit.eu-north-1.amazonaws.com	SHA256	b6KSK7xKq+V8j17iuA cjqXsG7zkqoUZZmmhY YFBq1wQ
git-codecommit.me-south-1.amazonaws.com	MD5	0e:39:28:56:d5:41: e6:8d:fa:81:45:37: fb:f3:cd:f7
git-codecommit.me-south-1.amazonaws.com	SHA256	0+NTToCGgjHekiBu01 0ad7R0GESz+DBLX0d/ c9wc0JU
git-codecommit.ap-east-1.amazonaws.com	MD5	a8:00:3d:24:52:9d: 61:0e:f6:e3:88:c8: 96:01:1c:fe

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.ap-east-1.amazonaws.com	SHA256	LafadYwUYW8h0NoTRp objbjNs9IRnbEwHtezD 3aAIBX0
git-codecommit.cn-north-1.amazonaws.com.cn	MD5	11:7e:2d:74:9e:3b: 94:a2:69:14:75:6f: 5e:22:3b:b3
git-codecommit.cn-north-1.amazonaws.com.cn	SHA256	IYUXxH20pTDsyYMLIp +JY8CTLS4UX+ZC5JVZ XPRaxc8
git-codecommit.cn-northwest-1.amazonaws.com.cn	MD5	2e:a7:fb:4c:33:ac: 6c:f9:aa:f2:bc:fb: 0a:7b:1e:b6
git-codecommit.cn-northwest-1.amazonaws.com.cn	SHA256	wqjd6eHd0+m0Bx+dCN uL0omUoCNjaDtZiEpW j5TmCfQ
git-codecommit.eu-south-1.amazonaws.com	MD5	b9:f6:5d:e2:48:92: 3f:a9:37:1e:c4:d0: 32:0e:fb:11
git-codecommit.eu-south-1.amazonaws.com	SHA256	1yXrWbCg3uQmJr11Xx B/ASR7ugW1Ysf5yzY0 JbudHsI
git-codecommit.ap-northeast-3.amazonaws.com	MD5	25:17:40:da:b9:d4: 18:c3:b6:b3:fb:ed: 1c:20:fe:29
git-codecommit.ap-northeast-3.amazonaws.com	SHA256	2B815B9F0AvwLnRxSV xUz4kDYmtEQUGGdQYP 8OQLXhA

Server	Tipo de hash criptográfico	Huella digital
git-codecommit.af-south-1.amazonaws.com	MD5	21:a0:ba:d7:c1:d1: b5:39:98:8d:4d:7c: 96:f5:ca:29
git-codecommit.af-south-1.amazonaws.com	SHA256	C34ji3x/cnsDZjUpyN GXdE5pjHYimqJrQZ3l eTgqJHM
git-codecommit.il-central-1.amazonaws.com	MD5	04:74:89:16:98:7a: 61:b1:69:46:42:3c: d1:b4:ac:a9
git-codecommit.il-central-1.amazonaws.com	SHA256	uFxhp51kUWhleTLeYb xQVYm4RnNLNZ5Dbdm1 cgdS1/8

Utilización AWS CodeCommit con puntos finales de VPC de interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y CodeCommit. Puede utilizar esta conexión para habilitar que CodeCommit se comunique con sus recursos en su VPC sin pasar por la red pública de Internet.

Amazon VPC es un AWS servicio que puede utilizar para lanzar AWS recursos en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de enrutamiento y las puertas de enlace de red. Con los puntos de enlace de la VPC, la AWS red gestiona el enrutamiento entre la VPC y AWS los servicios, y puede utilizar las políticas de IAM para controlar el acceso a los recursos del servicio.

Para conectar su VPC CodeCommit, debe definir un punto final de VPC de interfaz para CodeCommit. Un punto final de interfaz es una interfaz de red elástica con una dirección IP privada que sirve como punto de entrada para el tráfico destinado a un AWS servicio compatible. El punto final proporciona una conectividad fiable y escalable CodeCommit sin necesidad de una puerta de

enlace a Internet, una instancia de traducción de direcciones de red (NAT) o una conexión VPN. Para obtener más información, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC.

Note

Es posible que otros AWS servicios que proporcionan soporte de VPC y con los que se integran CodeCommit, por ejemplo, no admitan el uso de puntos de enlace de Amazon VPC para esa integración. AWS CodePipeline Por ejemplo, el tráfico entre CodePipeline y CodeCommit no se puede restringir al rango de subredes de la VPC. Los servicios que sí admiten la integración, por ejemplo [AWS Cloud9](#), pueden requerir servicios adicionales, como AWS Systems Manager.

Los puntos finales de VPC de interfaz cuentan con una AWS tecnología que permite la comunicación privada entre AWS servicios mediante una interfaz de red elástica con direcciones IP privadas. AWS PrivateLink Para obtener más información, consulte [AWS PrivateLink](#).

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulte [Introducción](#) en la Guía del usuario de Amazon VPC.

Disponibilidad.

CodeCommit actualmente es compatible con los puntos finales de VPC de las siguientes maneras:

Regiones de AWS

- Este de EE. UU. (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- EE.UU. Oeste (Oregón)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Europa (Fráncfort)
- Europa (Estocolmo)
- Europa (Milán)
- África (Ciudad del Cabo)

- Israel (Tel Aviv)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Yakarta)
- Medio Oriente (EAU)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Mumbai)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Hong Kong)
- América del Sur (São Paulo)
- Middle East (Bahrain)
- Canadá (centro)
- China (Pekín)
- China (Ningxia)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

Cree puntos finales de VPC para CodeCommit

Para empezar a usarlo CodeCommit con su VPC, cree un punto de enlace de VPC de interfaz para. CodeCommit CodeCommit requiere puntos finales separados para las operaciones de Git y para las operaciones CodeCommit de API. En función de sus necesidades empresariales, es posible que tenga que crear más de un punto de conexión de VPC. Al crear un punto final de VPC para CodeCommit, elija AWS Servicios y, en Nombre del servicio, elija una de las siguientes opciones:

- `com.amazonaws. region.git-codecommit`: elige esta opción si quieres crear un punto final de VPC para las operaciones de Git con repositorios. CodeCommit Por ejemplo, elige esta opción si tus usuarios utilizan un cliente Git y comandos como `git pullgit commit`, y `git push` cuando interactúan con CodeCommit los repositorios.
- `com.amazonaws. region.git-codecommit-fips`: elige esta opción si quieres crear un punto final de VPC para las operaciones de Git con CodeCommit repositorios que cumpla con la publicación

140-2 del estándar federal de procesamiento de información (FIPS), norma gubernamental estadounidense.

Note

Los puntos de conexión FIPS para Git no están disponibles en todas las AWS regiones. Para obtener más información, consulte [Puntos de conexión de Git](#).

- `com.amazonaws.region.codecommit`: elige esta opción si quieres crear un punto final de VPC para las operaciones de la API. CodeCommit Por ejemplo, elige esta opción si tus usuarios utilizan la AWS CLI CodeCommit API o los AWS SDK CodeCommit para interactuar con ellos en operaciones como `CreateRepository`, y. `ListRepositories` `PutFile`
- `com.amazonaws.region.codecommit-fips`: elija esta opción si desea crear un punto final de VPC para las operaciones de la CodeCommit API que cumpla con la norma federal de procesamiento de información (FIPS), publicación 140-2, del gobierno de EE. UU.

Note

Los puntos de conexión FIPS no están disponibles en todas las regiones. AWS Para obtener más información, consulte la entrada correspondiente a la descripción general de AWS CodeCommit la [norma federal de procesamiento de información \(FIPS\) 140-2](#).

Cree una política de puntos de conexión de VPC para CodeCommit

Puede crear una política para los puntos de enlace de Amazon VPC CodeCommit en la que pueda especificar:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden realizar acciones.

Por ejemplo, una empresa podría querer restringir el acceso a repositorios al rango de direcciones de red para una VPC. Puede ver un ejemplo de este tipo de política aquí: [Ejemplo 3: permitir a un usuario conectarse desde un rango de direcciones IP específicas para obtener acceso a un repositorio](#). La empresa configuró dos puntos de conexión de VPC de Git para la región este de EE. UU. (Ohio): `com.amazonaws.us-east-2.codecommit` y `com.amazonaws.us-`

east-2.git-codecommit-fips. Quieren permitir el envío de código a un CodeCommit repositorio cuyo nombre *MyDemoRepo* solo esté en el FIPS-compliant punto final. Para obligar a ello, configurarían una política similar a la siguiente en el punto de conexión de `com.amazonaws.us-east-2.codecommit` que deniega específicamente las acciones de inserción de Git:

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "codecommit:GitPush",
      "Effect": "Deny",
      "Resource": "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
      "Principal": "*"
    }
  ]
}
```

Important

La clave de condición global no `aws:VpcSourceIp` es compatible con CodeCommit los repositorios de las políticas de IAM para los comandos `git push`

Para obtener más información, consulte [Creación de un punto de conexión de tipo interfaz](#) en la Guía del usuario de Amazon VPC.

Cuotas en AWS CodeCommit

En la siguiente tabla se describen las cuotas en CodeCommit. Para obtener información acerca de las cuotas que se pueden cambiar, consulte [Cuotas y puntos de enlace de AWS CodeCommit](#). Para obtener información acerca de cómo solicitar un aumento de cuotas de servicio, consulte [AWS Service Quotas](#). Para obtener información sobre las versiones requeridas de Git y otro software, consulte [Compatibilidad entre CodeCommit, Git y otros componentes](#).

Nombres de reglas de aprobación y plantillas de reglas de aprobación	Cualquier combinación de letras, números, puntos, espacios, guiones bajos y guiones de entre 1 y 100 caracteres. Los nombres distinguen mayúsculas de minúsculas. Los nombres de repositorios no pueden terminar en .git ni pueden contener los siguientes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Longitud del contenido de la regla de aprobación	3.000 caracteres
Longitud de la descripción de la plantilla de regla de aprobación	1.000 caracteres
Referencias de destino de plantilla de regla de aprobación	100
Plantillas de reglas de aprobación	1000 en un Región de AWS
Reglas de aprobación en una solicitud de extracción	Hasta 30 como máximo. Hasta 25 de ellos pueden proceder de plantillas de reglas de aprobación.
Reglas de aprobación de una solicitud de extracción creadas a partir de una plantilla de regla de aprobación	25
Aprobaciones de una solicitud de extracción	200
Aprobadores de un grupo de aprobación	50
Nombres de ramificaciones	Cualquier combinación de caracteres permitida de entre 1 y 256 caracteres de longitud, excepto los nombres de ramificación que contengan exactamente 40 caracteres hexadecimales. Los nombres de ramificaciones no pueden:

	• 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: ? ^ * [\ ~ : 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 Cuestiones internas de Git y git-check-ref-format.
Longitud de comentario	Máximo de 10 240 caracteres.
Tamaño de archivo comprimido para inicializar un CodeCommit repositorio en AWS CloudFormation	Máximo de 20 GB cuando está comprimido.
Datos personalizados para disparadores	Es un campo de cadena limitado a 1 000 caracteres. No se puede usar para transmitir parámetros dinámicos.
Visualizar en la consola	Es posible que un archivo o una comparación entre archivos no se puedan visualizar en la consola si: • El archivo pesa más de 2 MB • El archivo contiene más de 25 000 caracteres en una sola línea • Una comparación contiene más de 6500 líneas de diferencias en total

Direcciones de correo electrónico en confirmaciones realizadas en la consola

Cualquier combinación de caracteres permitidos de entre 1 y 256 caracteres. Las direcciones de correo electrónico no se validan.

Rutas de archivo

Cualquier combinación de caracteres permitidos de entre 1 y 4096 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:

- contener cadenas vacías;
- ser una ruta de archivo relativa;
- incluir ninguna de las siguientes combinaciones de caracteres:

`./`

`../`

`//`

- terminar con una barra diagonal final o barra inversa.

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 CodeCommit repositorio, utilice los estándares de Amazon Linux.

Tamaño del archivo

Un máximo de 6 MB para cada archivo individual cuando utilice la CodeCommit consola, las API o el AWS CLI.

Tamaño de blob de Git	Máximo de 2 GB.  Note 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.
Visualización gráfica de ramificaciones en Commit Visualizer	35 por página. Si hay más de 35 ramificaciones en una única página, el gráfico no se muestra.
Metadatos para confirmaciones	Un máximo de 20 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) cuando se utilice la CodeCommit consola, las API o el AWS CLI.  Note No hay límite al número ni al tamaño total de archivos en una única confirmación, siempre que los datos no superen los 6 MB y que ningún archivo individual supere los 6 MB y ningún blob supere los 2 GB por sí solo.
Número de archivos en una confirmación	Máximo de 100.
Número de solicitudes de extracción abiertas	Máximo de 1000.
Número de referencias en una única inserción	Máximo de 4000, incluidas las de creación, eliminación y actualización. No hay límite para el número total de referencias en el repositorio.

Número de repositorios	Un máximo de 5000 por cuenta de Amazon Web Services. Este límite puede cambiarse. Para obtener más información, consulte puntos de conexión de AWS CodeCommit y cuotas y Service Quotas de AWS .
Número de disparadores en un repositorio	Máximo de 10.

Regions

CodeCommit está disponible en las siguientes ubicaciones Regiones de AWS:

- Este de EE. UU. (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- EE.UU. Oeste (Oregón)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Europa (Fráncfort)
- Europa (Estocolmo)
- Europa (Milán)
- África (Ciudad del Cabo)
- Israel (Tel Aviv)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sidney)
- Asia-Pacífico (Yakarta)
- Medio Oriente (EAU)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Mumbai)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Hong Kong)
- América del Sur (São Paulo)
- Middle East (Bahrain)
- Canadá (centro)
- China (Pekín)
- China (Ningxia)
- AWS GovCloud (US-West)

	• AWS GovCloud (US-East) Para obtener más información, consulte Regiones y puntos de conexión de Git.
Descripciones de repositorios	Cualquier combinación de caracteres de entre 0 y 1000 caracteres. Las descripciones de repositorios son opcionales.
Nombres de repositorio	Cualquier combinación de letras, números, puntos, guiones bajos y guiones de entre 1 y 100 caracteres. Los nombres distinguen mayúsculas de minúsculas. Los nombres de repositorios no pueden terminar en .git y no pueden contener los siguientes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Nombres de clave de etiqueta de repositorio	Cualquier combinación de letras, números, espacios y caracteres permitidos de Unicode con una longitud de UTF-8 entre 1 y 128 caracteres. Los caracteres permitidos son + - = . _ : / @ Los nombres de clave de etiqueta deben ser únicos y cada clave solo puede tener un valor. Una etiqueta no puede: • comenzar por aws : • contener únicamente espacios • terminar con un espacio • contener emojis ni ninguno de los siguientes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;

Valores de etiqueta de repositorio	Cualquier combinación de letras Unicode, números, espacios y caracteres permitidos de UTF-8 entre 1 y 256 caracteres de longitud. Los caracteres permitidos son + - = . _ : / @ Una clave solo puede tener un valor, pero muchas claves pueden tener el mismo valor. Una etiqueta no puede: • contener únicamente espacios • terminar con un espacio • contener emojis o alguno de los siguientes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Etiquetas del repositorio	Las etiquetas distinguen entre mayúsculas y minúsculas. Máximo de 50 por recurso. No se permiten etiquetas con nombres de 40 caracteres hexadecimales exactos.
Nombres de disparadores	Cualquier combinación de letras, números, puntos, guiones bajos y guiones de entre 1 y 100 caracteres. Los nombres de disparadores no pueden contener espacios ni comas.
Nombres de usuario en confirmaciones realizadas en la consola	Cualquier combinación de caracteres permitidos de entre 1 y 1,024 caracteres.

AWS CodeCommit referencia de línea de comandos

Esta información le ayuda a aprender a utilizar la AWS CLI.

Para instalar y configurar el AWS CLI

1. En su máquina local, descargue e instale el AWS CLI. Este es un requisito previo para interactuar con el CodeCommit desde la línea de comandos. Le recomendamos que instale la

versión 2 de AWS CLI . Es la versión principal más reciente AWS CLI y es compatible con todas las funciones más recientes. Es la única versión de la AWS CLI que admite el uso de una cuenta raíz, un acceso federado o credenciales temporales con `git-remote-codecommit`.

Para obtener más información, consulte [Cómo configurar la interfaz de línea de AWS comandos](#).

 Note

CodeCommit solo funciona con AWS CLI las versiones 1.7.38 y posteriores. Como práctica recomendada, instale o actualice AWS CLI a la última versión disponible. Para determinar qué versión de la AWS CLI ha instalado, ejecute el `aws --version` comando. Para actualizar una versión anterior AWS CLI a la versión más reciente, consulte [Instalación del AWS Command Line Interface](#).

2. Ejecute este comando para comprobar que los CodeCommit comandos de AWS CLI están instalados.

```
aws codecommit help
```

Este comando devuelve una lista de CodeCommit comandos.

3. Configure el AWS CLI con un perfil mediante el `configure` comando, de la siguiente manera:.

```
aws configure
```

Cuando se le solicite, especifique la clave de AWS acceso y la clave de acceso AWS secreta del usuario de IAM con CodeCommit el que va a utilizarlas. Además, asegúrese de especificar Región de AWS dónde se encuentra el repositorio, por ejemplo `us-east-2`. Cuando se le pregunte el formato de salida predeterminado, indique `json`. Por ejemplo, si está configurando un perfil para un usuario de IAM:

```
AWS Access Key ID [None]: Type your IAM user AWS access key ID here, and then press Enter
AWS Secret Access Key [None]: Type your IAM user AWS secret access key here, and then press Enter
Default region name [None]: Type a supported region for CodeCommit here, and then press Enter
Default output format [None]: Type json here, and then press Enter
```

Para obtener más información sobre la creación y configuración de perfiles para utilizarlos con el AWS CLI, consulte lo siguiente:

- [Perfiles con nombre](#)
- [Uso de un rol de IAM en el AWS CLI](#)
- [Comando Set](#)
- [Conexión a AWS CodeCommit repositorios con credenciales rotativas](#)

Para conectarse a un repositorio o un recurso de otro Región de AWS, debe volver a configurarlo AWS CLI con el nombre de región predeterminado. Los nombres de región predeterminados admitidos CodeCommit incluyen:

- us-east-2
- us-east-1
- eu-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- ap-southeast-3
- me-central-1
- eu-central-1
- ap-northeast-2
- sa-east-1
- us-west-1
- eu-west-2
- ap-south-1
- ap-south-1
- ca-central-1
- us-gov-west-1
- us-gov-east-1

- ap-east-1
- me-south-1
- cn-north-1
- cn-northwest-1
- eu-south-1
- ap-northeast-3
- af-south-1
- il-central-1

Para obtener más información sobre CodeCommit y Región de AWS, consulte [Regiones y puntos de conexión de Git](#). Para obtener más información sobre IAM, claves de acceso y claves secretas, consulte [¿Cómo puedo obtener credenciales?](#) y [Gestión de claves de acceso para usuarios de IAM](#). Para obtener más información sobre los perfiles AWS CLI y, consulte [Perfiles con nombre](#).

Para ver una lista de todos los CodeCommit comandos disponibles, ejecute el siguiente comando:

```
aws codecommit help
```

Para ver información sobre un CodeCommit comando, ejecute el siguiente comando, donde *command-name* aparece el nombre del comando (por ejemplo, create-repository):

```
aws codecommit command-name help
```

Consulte las siguientes descripciones y ejemplos de uso de los comandos en el AWS CLI:

- [associate-approval-rule-template-with-repository](#)
- [batch-associate-approval-rule-template-with-repositories](#)
- [batch-disassociate-approval-rule-template-from-repositories](#)
- [batch-describe-merge-conflicts](#)
- [batch-get-commits](#)
- [batch-get-repositories](#)
- [create-approval-rule-template](#)

- [create-branch](#)
- [create-commit](#)
- [create-pull-request](#)
- [create-pull-request-approval-rule](#)
- [create-repository](#)
- [create-unreferenced-merge-commit](#)
- [delete-approval-rule-template](#)
- [delete-branch](#)
- [delete-comment-content](#)
- [delete-file](#)
- [delete-repository](#)
- [describe-merge-conflicts](#)
- [delete-pull-request-approval-rule](#)
- [describe-pull-request-events](#)
- [disassociate-pull-request-approval-rule-template-from-repository](#)
- [evaluate-pull-request-approval-rules](#)
- [get-approval-rule-template](#)
- [get-blob](#)
- [get-branch](#)
- [get-comment](#)
- [get-comment-reactions](#)
- [get-comments-for-compared-commit](#)
- [get-comments-for-pull-request](#)
- [get-commit](#)
- [get-differences](#)
- [get-merge-commit](#)
- [get-merge-conflicts](#)
- [get-merge-options](#)
- [get-pull-request](#)

- [get-pull-request-approval-states](#)
- [get-pull-request-override-state](#)
- [get-repository](#)
- [get-repository-triggers](#)
- [list-approval-rule-templates](#)
- [list-associated-approval-rule-templates-for-repository](#)
- [list-branches](#)
- [list-pull-requests](#)
- [list-repositories](#)
- [list-repositories-for-approval-rule-template](#)
- [list-tags-for-resource](#)
- [merge-branches-by-fast-forward](#)
- [merge-branches-by-squash](#)
- [merge-branches-by-three-way](#)
- [merge-pull-request-by-fast-forward](#)
- [merge-pull-request-by-squash](#)
- [merge-pull-request-by-three-way](#)
- [override-pull-request-approval-rules](#)
- [post-comment-for-compared-commit](#)
- [post-comment-for-pull-request](#)
- [post-comment-reply](#)
- [put-comment-reaction](#)
- [put-file](#)
- [put-repository-triggers](#)
- [tag-resource](#)
- [test-repository-triggers](#)
- [untag-resource](#)
- [update-approval-rule-template-content](#)
- [update-approval-rule-template-description](#)

- [update-approval-rule-template-name](#)
- [update-comment](#)
- [update-default-branch](#)
- [update-pull-request-approval-rule-content](#)
- [update-pull-request-approval-state](#)
- [update-pull-request-description](#)
- [update-pull-request-status](#)
- [update-pull-request-title](#)
- [update-repository-description](#)
- [update-repository-name](#)

Comandos básicos de Git

Puedes usar Git para trabajar con un repositorio local y el CodeCommit repositorio al que has conectado el repositorio local.

A continuación, se muestran algunos ejemplos básicos de los comandos de Git más utilizados.

Para ver más opciones, consulte la documentación de Git.

Temas

- [Variables de configuración](#)
- [Repositorios remotos](#)
- [Confirmaciones](#)
- [Ramificaciones](#)
- [Tags](#)

Variables de configuración

Enumera todas las variables de configuración.

```
git config --list
```

Muestra únicamente las variables de configuración locales.

```
git config --local -l
```

Muestra únicamente las variables de configuración del sistema.	<code>git config --system -l</code>
Muestra únicamente las variables de configuración globales.	<code>git config --global -l</code>
Establece una configuración variable en el archivo de configuración especificado.	<code>git config [--local --global --system] <i>variable-name</i> <i>variable-value</i></code>
Establece el nombre de la ramificación predeterminada como principal para todos los repositorios locales cuando se realiza una confirmación inicial en un repositorio que aún no tiene una ramificación predeterminada	<code>git config --global init.defaultBranch main</code>
Edita un archivo de configuración directamente. También se puede utilizar para detectar la ubicación de un archivo de configuración concreto. Para salir del modo de edición, normalmente escribe :q (para salir sin guardar los cambios) o :wq (para guardar los cambios y salir) y, a continuación, pulsa Intro.	<code>git config [--local --global --system] --edit</code>

Repositorios remotos

Inicializa un repositorio local para prepararlo para conectarlo a un repositorio. CodeCommit	<code>git init</code>
Se puede usar para configurar una conexión entre un repositorio local y un repositorio remoto (como un CodeCommit repositorio) utilizando el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y la URL especificada al repositorio. CodeCommit	<code>git remote add <i>remote-name</i> <i>remote-url</i></code>

Crea un repositorio local haciendo una copia de un CodeCommit repositorio en la URL especificada, en la subcarpeta especificada de la carpeta actual de la máquina local. Este comando también crea una rama de seguimiento remoto para cada rama del CodeCommit repositorio clonado y crea y extrae una rama inicial que se bifurca de la rama predeterminada actual del repositorio clonado. CodeCommit

```
git clone remote-url local-sub  
folder-name
```

Muestra el apodo que el repositorio local usa para el repositorio. CodeCommit

```
git remote
```

Muestra el apodo y la URL que el repositorio local usa para buscar y enviar archivos al repositorio. CodeCommit

```
git remote -v
```

Envía las confirmaciones finalizadas del repositorio local al repositorio, utilizando el apodo especificado que el CodeCommit repositorio local tiene para el repositorio y la rama especificada. CodeCommit También configura la información de seguimiento ascendente del repositorio local durante el envío.

```
git push -u remote-name branch-name
```

Envía las confirmaciones finalizadas del repositorio local al CodeCommit repositorio una vez configurada la información de seguimiento inicial.

```
git push
```

Extrae las confirmaciones finalizadas al repositorio local desde el CodeCommit repositorio, utilizando el apodo especificado que el repositorio local tiene para el repositorio y la rama especificada CodeCommit

```
git pull remote-name branch-name
```

Extrae del repositorio las confirmaciones finalizadas en el CodeCommit repositorio local una vez configurada la información de seguimiento inicial.

```
git pull
```

Desconecta el repositorio local del repositorio utilizando el apodo especificado que el CodeCommit repositorio local tiene para el repositorio. CodeCommit

```
git remote rm remote-name
```

Confirmaciones

Muestra lo que se ha añadido o no a la confirmación pendiente en el repositorio local.

```
git status
```

Muestra lo que se ha añadido o no a la confirmación pendiente en el repositorio local en un formato conciso.

```
git status -sb
```

(M = modificado, A= añadido, D= eliminado, etc.)

Muestra los cambios entre la confirmación pendiente y la última confirmación en el repositorio local.

```
git diff HEAD
```

Añade archivos específicos a la confirmación pendiente en el repositorio local.

```
git add [file-name-1 file-name-2  
file-name-N | file-pattern ]
```

Añade todos los archivos nuevos, modificados y eliminados a la confirmación pendiente en el repositorio local.

```
git add
```

Comienza la finalización de la confirmación pendiente en el repositorio local, que muestra un editor para proporcionar un mensaje de confirmación. Una vez que haya introducido el

```
git commit
```

mensaje, la confirmación pendiente se finalizará.	
Finaliza la confirmación pendiente en el repositorio local al mismo tiempo que especifica un mensaje de confirmación.	<code>git commit -m "<i>Some meaningful commit comment</i>"</code>
Enumera las confirmaciones recientes en el repositorio local.	<code>git log</code>
Muestra las confirmaciones recientes en el repositorio local en forma de gráfico.	<code>git log --graph</code>
Muestra las confirmaciones recientes en el repositorio local en un formato condensado predefinido.	<code>git log --pretty=oneline</code>
Muestra las confirmaciones recientes en el repositorio local en un formato condensado predefinido, con un gráfico.	<code>git log --graph --pretty=oneline</code>
Muestra las confirmaciones recientes en el repositorio local en un formato condensado personalizado, con un gráfico. (Para ver más opciones, consulte Entender el historial: confirmaciones en la documentación de Git).	<code>git log --graph --pretty=format:"%H (%h) : %cn : %ar : %s"</code>

Ramificaciones

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 CodeCommit repositorio 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 nueva ramificación en el repositorio local con el nombre de la ramificación especificada.	<code>git branch <i>new-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>
Crea una nueva ramificación en el repositorio local con el nombre de la ramificación especificada y, a continuación, cambia a ella.	<code>git checkout -b <i>new-branch-name</i></code>
Envía una nueva rama del repositorio local al CodeCommit repositorio con el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y el nombre de la rama especificada. También configura la información de seguimiento ascendente de la ramificación en el repositorio local durante el envío.	<code>git push -u <i>remote-name new-branch-name</i></code>

Crea una nueva ramificación en el repositorio local con el nombre de la ramificación especificada. A continuación, conecta la nueva rama del repositorio local a una rama existente en el CodeCommit repositorio, utilizando el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y el nombre de la rama especificada.

```
git branch --track new-branch-name  
remote-name /remote-branch-name
```

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 CodeCommit repositorio con el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y el nombre de la rama especificada. (Observe el uso de los dos puntos (:)).

```
git push remote-name :branch-name
```

Tags

Enumera todas las etiquetas del repositorio local.

```
git tag
```

Extrae todas las etiquetas del CodeCommit repositorio al repositorio local.

```
git fetch --tags
```

Muestra información sobre una etiqueta específica del repositorio local.

```
git show tag-name
```

Crea una etiqueta "ligera" en el repositorio local.

```
git tag tag-name commit-id-to-point-tag-at
```

Envía una etiqueta específica del repositorio local al CodeCommit repositorio utilizando el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y el nombre de la etiqueta especificada.

```
git push remote-name tag-name
```

Envía todas las etiquetas del repositorio local al repositorio con el apodo especificado que el CodeCommit repositorio local tiene para el repositorio. CodeCommit

```
git push remote-name --tags
```

Elimina una etiqueta del repositorio local.

```
git tag -d tag-name
```

Elimina una etiqueta del CodeCommit repositorio con el apodo especificado que el repositorio local tiene para el CodeCommit repositorio y el nombre de la etiqueta especificada. (Observe el uso de los dos puntos (:)).

```
git push remote-name :tag-name
```


AWS CodeCommit Historial de documentos de la Guía del usuario

En la siguiente tabla se describen los cambios importantes en la documentación de CodeCommit. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

- Versión de la API: 13/04/2015

Cambio	Descripción	Fecha
AWS CodeCommit ya está disponible para nuevos clientes.	AWS CodeCommit ya está disponible para nuevos clientes. Los clientes existentes de AWS CodeCommit pueden seguir utilizando el servicio con normalidad.	25 de noviembre de 2025
AWS CodeCommit ya no está disponible para nuevos clientes.	AWS CodeCommit ya no está disponible para nuevos clientes. Los clientes existentes de AWS CodeCommit pueden seguir utilizando el servicio con normalidad. Más información	25 de julio de 2024
Se ha actualizado la información de solución de problemas para el ayudante de credenciales	Se ha agregado información sobre la solución de errores causados por archivos .gitconfig incompletos o mal configurados.	9 de julio de 2024
Información de solución de problemas actualizada para git-remote-codecommit	Se ha agregado información adicional sobre la solución de errores en entornos administrados externamente.	9 de julio de 2024

[Se eliminaron las referencias a AWS Toolkit for Eclipse](#)

Ha AWS Toolkit for Eclipse llegado al final de su vida útil y ya no es compatible. Se han eliminado las referencias a él.

9 de julio de 2024

[CodeCommit ahora admite el uso de claves gestionadas por el cliente](#)

Ahora puede utilizar una clave gestionada por el cliente o una Clave administrada de AWS para cifrar y descifrar los datos de un repositorio. Para obtener más información, consulte [AWS KMS y cifrado](#), [Crear un repositorio](#) y [Cambiar la configuración del repositorio](#).

21 de diciembre de 2023

[CodeCommit ya está disponible en Israel \(Tel Aviv\);](#)

CodeCommit ya está disponible en Israel (Tel Aviv). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git](#).

28 de agosto de 2023

[Cambios en las políticas gestionadas para CodeCommit](#)

Las AWSCodeCommitFullAccess políticas AWSCodeCommitPowerUser y se han actualizado con un permiso adicional. Para obtener más información, consulta [CodeCommit las actualizaciones de las políticas AWS gestionadas](#).

16 de mayo de 2023

[CodeCommit ahora está disponible en tres versiones adicionales Regiones de AWS](#)

CodeCommit ahora está disponible en tres más Regiones de AWS: Asia Pacífico (Yakarta), Oriente Medio (Emiratos Árabes Unidos) y Asia Pacífico (Hyderabad). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git](#).

28 de febrero de 2023

[CodeCommit ya está disponible en África \(Ciudad del Cabo\)](#)

CodeCommit ahora está disponible en una versión adicional Región de AWS: África (Ciudad del Cabo). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git](#).

15 de septiembre de 2021

[Cambios en las políticas gestionadas para CodeCommit](#)

Los detalles sobre las actualizaciones de las políticas AWS administradas ya CodeCommit están disponibles. Para obtener más información, consulte [CodeCommit las actualizaciones de las políticas AWS administradas](#).

18 de agosto de 2021

[CodeCommit ya está disponible en Asia Pacífico \(Osaka\)](#)

CodeCommit ahora está disponible en una versión adicional Región de AWS: Asia-Pacífico (Osaka). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git](#).

14 de abril de 2021

[AWS CloudFormation y AWS Cloud Development Kit \(AWS CDK\) cambia el comportamiento de nomenclatura de las ramas predeterminadas en CodeCommit](#)

Los repositorios creados con CloudFormation o AWS CDK con una confirmación inicial de código ahora usan el nombre de rama predeterminado, main. Este cambio no afecta a los repositorios o ramificaciones existentes. Los clientes que utilizan clientes Git locales para crear sus confirmaciones iniciales tienen un nombre de ramificación predeterminado que sigue la configuración de esos clientes de Git. Para obtener más información, consulte [Crear CodeCommit recursos con AWS CloudFormation](#).

4 de marzo de 2021

[CodeCommit cambia el comportamiento de nomenclatura de las ramas predeterminadas](#)

A partir del 19 de enero de 2021, el nombre de rama predeterminado creado por una confirmación inicial en un CodeCommit repositorio es main. Este cambio no afecta a los repositorios o ramificaciones existentes. Los clientes que utilizan clientes Git locales para crear sus confirmaciones iniciales tienen un nombre de ramificación predeterminado que sigue la configuración de esos clientes de Git. Para obtener más información, consulte [Cómo trabajar con ramificaciones](#), [Crear una confirmación](#) y [Cambiar la configuración de una ramificación](#).

19 de enero de 2021

[CodeCommit ya está disponible en Europa \(Milán\)](#)

CodeCommit ahora está disponible en una versión adicional Región de AWS: Europa (Milán). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git](#).

16 de septiembre de 2020

[CodeCommit añade compatibilidad con las reacciones de los emojis a los comentarios](#)

CodeCommit ahora permite reaccionar con emojis a los comentarios de otros usuarios. Para obtener más información, consulte [Comentar una confirmación](#) y [Revisar una solicitud de extracción](#).

24 de junio de 2020

[CodeCommit ahora disponible en China \(Pekín\) y China \(Ningxia\)](#)

CodeCommit ahora está disponible en dos versiones adicionales Regiones de AWS: China (Pekín) y China (Ningxia). Para obtener más información, consulte [Regiones y puntos de enlace de conexión de Git.](#)

23 de abril de 2020

[CodeCommit añade soporte para git-remote-codecommit](#)

CodeCommit admite conexiones a CodeCommit repositorios a través de HTTPS con git-remote-codecommit, una utilidad que modifica Git. Este es el enfoque recomendado para las conexiones de acceso federado o temporal a los repositorios. CodeCommit También puede utilizar git-remote-codecommit con un usuario de IAM. git-remote-codecommit no requiere que configure las credenciales de Git para el usuario. Para obtener más información, consulte los [pasos de configuración de las conexiones HTTPS a AWS CodeCommit with.](#) git-remote-codecommit

4 de marzo de 2020

[CodeCommit admite etiquetas de sesión](#)

CodeCommit admite el uso de etiquetas de sesión, que son atributos de pares clave-valor que se transfieren cuando se asume una función de IAM, se utilizan credenciales temporales o se federa un usuario en (). AWS Security Token Service AWS STS Puede utilizar la información proporcionada en estas etiquetas para facilitar la identificación de quién hizo un cambio o causó un evento. Para obtener más información, consulte [Monitoreo de CodeCommit](#) y [Uso de etiquetas para proporcionar información de identidad en CodeCommit](#).

19 de diciembre de 2019

[CodeCommit está disponible en Asia Pacífico \(Hong Kong\)](#)

Ahora se puede utilizar CodeCommit en Asia Pacífico (Hong Kong). Para obtener más información, incluidos los puntos de enlace de conexión de Git, consulte [Regiones](#).

11 de diciembre de 2019

[CodeCommit es compatible con Amazon CodeGuru Reviewer](#)

CodeCommit es compatible con Amazon CodeGuru Reviewer, un servicio de revisión de código automatizado que utiliza el análisis de programas y el aprendizaje automático para detectar problemas comunes y recomendar soluciones en su código Java o Python. Para obtener más información, consulte [Asociar o desasociar un repositorio con Amazon CodeGuru Reviewer](#) y [Trabajar con solicitudes de extracción](#).

3 de diciembre de 2019

[CodeCommit admite las reglas de aprobación](#)

Ahora puede utilizar reglas de aprobación para ayudarle a personalizar los flujos de trabajo de desarrollo en los repositorios, de modo que las distintas ramificaciones tengan los niveles adecuados de aprobaciones y controles para las solicitudes de extracción. Para obtener más información, consulte [Trabajar con plantillas de reglas de aprobación](#) y [Trabajar con solicitudes de extracción](#).

20 de noviembre de 2019

[CodeCommit admite las reglas de notificación](#)

Ahora puede usar reglas de notificación para informar a los usuarios de los cambios importantes de los repositorios. La funcionalidad de esta función sustituye las notificaciones creadas antes del 5 de noviembre de 2019. Para obtener más información, consulte [Creación de una regla de notificación](#).

5 de noviembre de 2019

[CodeCommit está disponible en Oriente Medio \(Baréin\)](#)

Ahora se puede utilizar CodeCommit en Oriente Medio (Baréin). Para obtener más información, incluidos los puntos de enlace de conexión de Git, consulte [Regiones](#).

30 de octubre de 2019

[CodeCommit añade soporte para recuperar información sobre múltiples confirmaciones](#)

Puede obtener información sobre varias confirmaciones mediante el batch-get-commits comando de. AWS CLI Para obtener más información, consulte [Ver detalles de una confirmación](#).

15 de agosto de 2019

[CodeCommit está disponible en Europa \(Estocolmo\)](#)

Ahora se puede utilizar CodeCommit en Europa (Estocolmo). Para obtener más información, incluidos los puntos de enlace de conexión de Git, consulte [Regiones](#).

31 de julio de 2019

[CodeCommit añade soporte para etiquetar repositorios en la consola CodeCommit](#)

Ahora puede añadir, gestionar y eliminar etiquetas de un repositorio para ayudarle a gestionar sus AWS recursos desde la CodeCommit consola. Para obtener más información, consulte [Etiquetado de un repositorio](#).

2 de julio de 2019

[CodeCommit añade soporte para estrategias de fusión de Git adicionales](#)

A partir de ahora, puede elegir entre las estrategias de combinación de Git al combinar las solicitudes de extracción en CodeCommit. También puedes resolver los conflictos de fusión en la CodeCommit consola. Para obtener más información, consulte [Uso de solicitudes de extracción](#).

10 de junio de 2019

[CodeCommit está disponible en AWS GovCloud \(este de EE. UU.\)](#)

Ahora se puede usar CodeCommit en AWS GovCloud (EE. UU. al este). Para obtener más información, incluidos los puntos de enlace de conexión de Git, consulte [Regiones](#).

31 de mayo de 2019

[CodeCommit añade soporte para etiquetar repositorios](#)

Ahora puedes añadir, gestionar y eliminar etiquetas de un repositorio para ayudarte a gestionar tus AWS recursos. Para obtener más información, consulte [Etiquetado de un repositorio](#).

30 de mayo de 2019

Buscar recursos en la consola	A partir de ahora, puede buscar rápidamente sus recursos, como repositorios, proyectos de compilación, aplicaciones de implementación y canalizaciones. Elija Ir a recurso o pulse la tecla / y, a continuación, escriba el nombre del recurso. Para obtener más información, consulte Tutorial de CodeCommit .	14 de mayo de 2019
CodeCommit está disponible en AWS GovCloud (US-West)	Ahora se puede utilizar CodeCommit en AWS GovCloud (EE. UU.-Oeste). Para obtener más información, incluidos los puntos de enlace de conexión de Git, consulte Regiones .	18 de abril de 2019
CodeCommit añade soporte para puntos de conexión de Amazon VPC	Ahora puede establecer una conexión privada entre su VPC y. CodeCommit Para obtener más información, consulte Uso CodeCommit con puntos finales de VPC de interfaz .	7 de marzo de 2019
CodeCommit añade una nueva API	CodeCommit ha añadido una API para crear confirmaciones. Para obtener más información, consulte la sección Creación de una confirmación .	20 de febrero de 2019

[Actualización del contenido](#)

El contenido de esta guía se ha actualizado con pequeñas correcciones y orientación adicional para la resolución de problemas.

2 de enero de 2019

[Actualización del contenido](#)

El contenido de esta guía se ha actualizado para adaptarlo a la nueva experiencia de CodeCommit consola.

30 de octubre de 2018

[CodeCommit y el estándar federal de procesamiento de información \(FIPS\)](#)

CodeCommit ha añadido compatibilidad con la publicación 140-2 del Estándar Federal de Procesamiento de Información (FIPS), norma gubernamental en algunas regiones. Para obtener más información acerca de los puntos de enlace de FIPS y FIPS, consulte la [descripción general del Estándar de Procesamiento de la Información Federal \(FIPS\) 140-2](#). Para obtener más información sobre los puntos de enlace de conexión de Git, consulte [Regiones](#).

25 de octubre de 2018

[CodeCommit suma tres APIs](#)

CodeCommit ha añadido tres APIs para facilitar el trabajo con archivos. Para obtener más información acerca de los puntos de conexión de Git, consulte [Permisos para realizar acciones en archivos individuales](#) y la [Referencia de API de AWS CodeCommit](#).

27 de septiembre de 2018

[CodeCommit notificación del historial de documentación disponible a través de la fuente RSS](#)

Ahora puede recibir notificaciones sobre las actualizaciones de la CodeCommit documentación suscribiéndose a una fuente RSS.

29 de junio de 2018

Actualizaciones anteriores

En la siguiente tabla se describen cambios importantes en la documentación anteriores al 29 de junio de 2018.

Cambio	Descripción	Fecha de modificación
Nuevo tema	Se ha agregado el tema Limitación de combinaciones e inserciones en ramificaciones . Se ha actualizado el tema CodeCommit referencia de permisos .	16 de mayo de 2018
Sección nueva	Se ha agregado la sección Trabajar con archivos en repositorios de AWS CodeCommit . Se han actualizado los temas CodeCommit referencia de permisos y Empezar con AWS CodeCommit .	21 de febrero de 2018
Nuevo tema	Se ha agregado el tema Configura el acceso multicuenta a un AWS CodeCommit repositorio mediante roles .	21 de febrero de 2018

Cambio	Descripción	Fecha de modificación
Nuevo tema	Se ha agregado el tema Integre AWS Cloud9 con AWS CodeCommit . El Integraciones de productos y servicios tema se ha actualizado con información sobre AWS Cloud9.	1 de diciembre de 2017
Sección nueva	Se ha agregado la sección Trabajar con solicitudes de cambios en AWS CodeCommit repositorios . La sección Autenticación y control de acceso para AWS CodeCommit se ha actualizado con información sobre los permisos para las solicitudes de extracción y los comentarios. También incluye declaraciones actualizadas de políticas administradas.	20 de noviembre de 2017
Temas actualizados	El Integraciones de productos y servicios tema se ha actualizado para incluir enlaces para los clientes que desean actualizar sus canalizaciones existentes para usar Amazon CloudWatch Events para iniciar canalizaciones en respuesta a los cambios en un CodeCommit repositorio.	11 de octubre de 2017
Temas nuevos	Se ha agregado la sección Autenticación y control de acceso para AWS CodeCommit . Sustituye al tema Referencia sobre los permisos de acceso.	11 de septiembre de 2017
Temas actualizados	La sección Administración de los disparadores de un repositorio se ha actualizado para reflejar los cambios en la configuración de los disparadores. Se han actualizado los temas y las imágenes en toda la guía para reflejar los cambios en la barra de navegación.	29 de agosto de 2017
Nuevo tema	Se ha agregado el tema Trabajo con las preferencias del usuario . Se ha actualizado el tema Visualización de información detallada de las etiquetas . Los Integraciones de productos y servicios temas se han actualizado con información sobre la integración con Amazon CloudWatch Events.	3 de agosto de 2017

Cambio	Descripción	Fecha de modificación
Temas nuevos	Se han agregado los Integrar Visual Studio con AWS CodeCommit temas Integrar Eclipse con AWS CodeCommit y.	29 de junio de 2017
Tema actualizado	CodeCommit ahora está disponible en dos regiones adicionales: Asia Pacífico (Bombay) y Canadá (Central). Se ha actualizado el tema Regiones y puntos de conexión de Git .	29 de junio de 2017
Tema actualizado	CodeCommit ahora está disponible en cuatro regiones adicionales: Asia Pacífico (Seúl), Sudamérica (São Paulo), EE.UU. Oeste (Norte de California) y Europa (Londres). Se ha actualizado el tema Regiones y puntos de conexión de Git .	6 de junio de 2017
Tema actualizado	CodeCommit ahora está disponible en cuatro regiones adicionales: Asia Pacífico (Tokio), Asia Pacífico (Singapur), Asia Pacífico (Sídney) y Europa (Fráncfort). El Regiones y puntos de conexión de Git tema se ha actualizado para proporcionar información sobre los puntos finales de conexión de Git y las regiones compatibles. CodeCommit	25 de mayo de 2017
Nuevo tema	Se ha agregado el tema Comparación y combinación de ramificaciones . El contenido de la sección Uso de las ramificaciones se ha actualizado con información acerca de cómo usar la consola de CodeCommit para trabajar con ramificaciones en un repositorio.	18 de mayo de 2017
Nuevo tema	Se ha agregado el tema Comparar confirmaciones con información sobre la comparación de confirmaciones. La estructura de la guía de usuario se ha actualizado para trabajar con repositorios , confirmaciones , y ramificaciones .	28 de marzo de 2017

Cambio	Descripción	Fecha de modificación
Tema actualizado	El tema Ver detalles de una confirmación se ha actualizado con información sobre cómo ver la diferencia entre una confirmación y su elemento primario en la consola y el uso del comando <code>get-differences</code> para ver las diferencias entre confirmaciones utilizando la AWS CLI.	24 de enero de 2017
Nuevo tema	Se ha añadido al Registrar llamadas a la AWS CodeCommit API con AWS CloudTrail tema información sobre cómo registrar las conexiones y CodeCommit AWS CloudFormation utilizarlas.	11 de enero de 2017
Nuevo tema	El Para usuarios HTTPS mediante credenciales de Git tema se ha agregado con información sobre cómo configurar las conexiones para CodeCommit usar las credenciales de Git a través de HTTPS.	22 de diciembre de 2016
Tema actualizado	El Integraciones de productos y servicios tema se ha actualizado para incluir información sobre la integración con AWS CodeBuild.	5 de diciembre de 2016
Tema actualizado	CodeCommit ahora está disponible en otra región, Europa (Irlanda). El Regiones y puntos de conexión de Git tema se ha actualizado para proporcionar información sobre los puntos finales de conexión de Git y las regiones compatibles. CodeCommit	16 de noviembre de 2016
Tema actualizado	CodeCommit ya está disponible en otra región, el oeste de EE. UU. (Oregón). El Regiones y puntos de conexión de Git tema se ha actualizado para proporcionar información sobre los puntos finales de conexión de Git y las regiones compatibles. CodeCommit	14 de noviembre de 2016

Cambio	Descripción	Fecha de modificación
Nuevo tema	El Creación de un disparador para una función de Lambda tema se ha actualizado para reflejar la capacidad de crear CodeCommit activadores como parte de la creación de la función Lambda. Este proceso simplificado agiliza la creación de activadores y configura automáticamente el desencadenador con los permisos necesarios para CodeCommit invocar la función Lambda. El Crear un disparador para una función de Lambda existente tema se ha agregado para incluir información sobre la creación de activadores para las funciones Lambda existentes en la CodeCommit consola.	19 de octubre de 2016
Nuevo tema	CodeCommit ahora está disponible en otra región, EE. UU. Este (Ohio). El Regiones y puntos de conexión de Git tema se ha agregado para proporcionar información sobre los puntos finales de conexión de Git y las regiones compatibles. CodeCommit	17 de octubre de 2016
Actualización de tema	El Integraciones de productos y servicios tema se ha actualizado para incluir información sobre la integración con AWS Elastic Beanstalk.	13 de octubre de 2016
Actualización de tema	El Integraciones de productos y servicios tema se ha actualizado para incluir información sobre la integración con CloudFormation.	6 de octubre de 2016
Actualización de tema	Se ha revisado el tema Para conexiones SSH en Windows para proporcionar orientación para utilizar un emulador de Bash para las conexiones SSH en Windows, en lugar de la suite de herramientas PuTTY.	29 de septiembre de 2016

Cambio	Descripción	Fecha de modificación
Actualización de tema	Los Empezando con CodeCommit temas Ver detalles de una confirmación y se han actualizado para incluir información sobre el visualizador de confirmaciones en la CodeCommit consola. Se ha actualizado el tema Cuotas con el aumento del número de referencias permitidas en una única inserción.	14 de septiembre de 2016
Actualización de tema	Los Empezando con CodeCommit temas Ver detalles de una confirmación y se han actualizado para incluir información sobre cómo ver el historial de confirmaciones en la CodeCommit consola.	28 de julio de 2016
Temas nuevos	Se han agregado los temas Migrar un repositorio de Git a AWS CodeCommit y Migre contenido local o no versionado a AWS CodeCommit .	29 de junio de 2016
Actualización de tema	Se han realizado actualizaciones menores en los temas Solución de problemas y Para las conexiones HTTPS en Windows con el ayudante de credenciales AWS CLI .	22 de junio de 2016
Actualización de tema	Los temas Integraciones de productos y servicios y la referencia sobre los permisos de acceso se han actualizado para incluir información sobre la integración con CodePipeline.	18 de abril de 2016
Temas nuevos	Se ha agregado la sección Administración de los disparadores de un repositorio . Los nuevos temas incluyen ejemplos, entre otros de políticas y muestras de código, sobre cómo crear, editar y eliminar disparadores.	7 de marzo de 2016
Nuevo tema	Se ha agregado el tema Integraciones de productos y servicios . Se han realizado actualizaciones menores a Solución de problemas .	7 de marzo de 2016

Cambio	Descripción	Fecha de modificación
Actualización de tema	Además de la huella digital MD5 del servidor, se CodeCommit ha agregado la huella digital SHA256 del servidor para Para conexiones SSH en Linux, macOS o Unix y Para conexiones SSH en Windows .	9 de diciembre de 2015
Nuevo tema	Se ha agregado el tema Navegación por los archivos en un repositorio . Se han agregado nuevas cuestiones a Solución de problemas . Se han realizado mejoras y correcciones menores en toda la guía del usuario.	5 de octubre de 2015
Nuevo tema	Se ha agregado el tema Para los usuarios de SSH que no utilizan AWS CLI . Se han racionalizado los temas de la sección Configuración . Se han facilitado instrucciones para ayudar a los usuarios a determinar qué pasos han de seguir para sus sistemas operativos y protocolos de preferencia.	5 de agosto de 2015
Actualización de tema	Se han añadido aclaraciones y ejemplos a los pasos de la ID de clave SSH en SSH y Linux, macOS o Unix: configura las claves públicas y privadas para Git y CodeCommit y Paso 3: Configura las claves públicas y privadas para Git y CodeCommit .	24 de julio de 2015
Actualización de tema	Se han actualizado los pasos de Paso 3: Configura las claves públicas y privadas para Git y CodeCommit para abordar un problema con IAM y el guardado del archivo de clave pública.	22 de julio de 2015
Actualización de tema	Se ha actualizado Solución de problemas con ayudas a la navegación. Se ha agregado más información para la solución de problemas con las cadenas de credenciales.	20 de julio de 2015
Actualización de tema	Se ha agregado más información sobre AWS Key Management Service los permisos a los temas AWS KMS y cifrado y a la referencia de permisos de acceso.	17 de julio de 2015

Cambio	Descripción	Fecha de modificación
Actualización de tema	Se ha agregado otra sección a Solución de problemas con información sobre la solución de problemas con AWS Key Management Service.	10 de julio de 2015
Versión inicial	Esta es la versión inicial de la Guía de usuario de CodeCommit .	9 de julio de 2015

AWSGlosario de

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.