

Guida per l'utente

AWS CodeCommit



Versione API 2015-04-13

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

AWS CodeCommit: Guida per l'utente

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è CodeCommit?	1
Presentazione CodeCommit	2
CodeCommit, Git e la scelta del AWS servizio giusto per le tue esigenze	3
Come funziona? CodeCommit	6
In cosa CodeCommit differisce il controllo delle versioni dei file in Amazon S3?	8
Come posso iniziare? CodeCommit	8
Dove posso saperne di più su Git?	8
Configurazione	9
Visualizza e gestisci le tue credenziali	9
Configurazione tramite credenziali Git	10
Configurazione con altri metodi	10
Compatibilità per CodeCommit, Git e altri componenti	12
Per utenti HTTPS che utilizzano credenziali Git	13
Fase 1: Configurazione iniziale per CodeCommit	13
Fase 2: installazione di Git	15
Passaggio 3: Creare credenziali Git per le connessioni HTTPS a CodeCommit	15
Passaggio 4: Connect alla CodeCommit console e clona il repository	17
Passaggi successivi	19
Per connessioni HTTPS con git-remote-codecommit	19
Fase 0: Installare i prerequisiti per git-remote-codecommit	20
Fase 1: Configurazione iniziale per CodeCommit	21
Fase 2: Installazione git-remote-codecommit	25
Passaggio 3: Connect alla CodeCommit console e clona il repository	26
Passaggi successivi	27
Per connessioni da strumenti di sviluppo	27
Integrazione AWS Cloud9 con AWS CodeCommit	30
Integra Visual Studio con AWS CodeCommit	34
Per gli utenti SSH che non utilizzano il AWS CLI	35
Passaggio 1: associa la tua chiave pubblica al tuo utente IAM	36
Passaggio 2: aggiungi CodeCommit alla tua configurazione SSH	37
Fasi successive	38
Per connessioni SSH su Linux, macOS o Unix	38
Fase 1: Configurazione iniziale per CodeCommit	38
Fase 2: installazione di Git	40

Passaggio 3: configurare le credenziali su Linux, macOS o Unix	40
Passaggio 4: Connect alla CodeCommit console e clona il repository	44
Fasi successive	46
Per le connessioni SSH su Windows	46
Fase 1: Configurazione iniziale per CodeCommit	47
Fase 2: installazione di Git	48
Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit	48
Passaggio 4: Connect alla CodeCommit console e clona il repository	53
Fasi successive	54
Per connessioni HTTPS su Linux, macOS o Unix con l'helper delle credenziali AWS CLI	54
Fase 1: Configurazione iniziale per CodeCommit	55
Fase 2: installazione di Git	59
Passaggio 3: configura l'helper delle credenziali	59
Passaggio 4: Connect alla CodeCommit console e clona il repository	61
Passaggi successivi	62
Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI	63
Fase 1: Configurazione iniziale per CodeCommit	64
Fase 2: installazione di Git	67
Passaggio 3: configura l'helper delle credenziali	68
Passaggio 4: Connect alla CodeCommit console e clona il repository	70
Passaggi successivi	71
Nozioni di base	72
Iniziare con CodeCommit	72
Prerequisiti	73
Fase 1: creazione di un repository CodeCommit	74
Passaggio 2: aggiungi file al tuo repository	77
Passaggio 3: sfoglia il contenuto del tuo repository	79
Passaggio 4: Crea e collabora su una pull request	84
Fase 5: rimozione	91
Fase 6: fasi successive	92
Guida introduttiva a Git e CodeCommit	92
Fase 1: Creare un repository CodeCommit	93
Fase 2: Creare un repository locale	95
Passaggio 3: crea il tuo primo commit	97
Passaggio 4: invia il tuo primo commit	98
Passaggio 5: condividi il CodeCommit repository e invia ed estrai un altro commit	98

Passaggio 6: Crea e condividi un ramo	101
Passaggio 7: Crea e condividi un tag	103
Passaggio 8: configurare le autorizzazioni di accesso	104
Fase 9: Pulizia	108
Integrazioni di prodotti e servizi	110
Integrazione con altri AWS servizi	110
Esempi di integrazione dalla community	117
Post del blog	117
Esempi di codice	121
Lavorare con i repository	122
Creazione di un repository	123
Crea un repository (console)	124
Crea un repository (AWS CLI)	125
Connessione a un repository	128
Prerequisiti per la connessione a un repository CodeCommit	128
Connect al CodeCommit repository clonando il repository	129
Connect un repository locale al CodeCommit repository	131
Condividi un repository	132
Scegli il protocollo di connessione da condividere con i tuoi utenti	133
Crea policy IAM per il tuo repository	134
Crea un gruppo IAM per gli utenti del repository	136
Condividi le informazioni di connessione con i tuoi utenti	137
Configurazione delle notifiche per gli eventi del repository	138
Utilizzo delle regole di notifica del repository	140
Creazione di una regola di notifica	140
Modificare o disabilitare le notifiche	144
Eliminare le notifiche	145
Tagging di un repository	146
Aggiunta di un tag a un repository	147
Visualizzazione dei tag per un repository	149
Modifica dei tag per un repository	150
Rimuovere un tag da un repository	152
Gestione dei trigger per un repository	153
Crea la risorsa e aggiungi le autorizzazioni per CodeCommit	154
Creare un trigger per un argomento di Amazon SNS	155
Creare un trigger per una funzione Lambda	162

Creare un trigger per una funzione Lambda esistente	167
Modifica i trigger per un repository	175
Trigger di test per un repository	177
Eliminare i trigger da un repository	179
Associare o dissociare un repository con Amazon Reviewer CodeGuru	182
Associa un repository a Reviewer CodeGuru	184
Annulla l'associazione di un repository da Reviewer CodeGuru	185
Visualizza i dettagli del repository	185
Visualizza i dettagli del repository (console)	186
Visualizza i dettagli CodeCommit del repository (Git)	186
Visualizza i dettagli CodeCommit del repository ()AWS CLI	188
Modifica delle impostazioni del repository	192
Modifica le impostazioni del repository (console)	192
Modifica le impostazioni AWS CodeCommit del repository ()AWS CLI	194
Sincronizza le modifiche tra i repository	196
Invia i commit a due repository	197
Configura l'accesso tra account a un repository utilizzando i ruoli	202
Accesso all'archivio su più account: azioni per l'amministratore in AccountA	203
Accesso all'archivio tra account: azioni per l'amministratore in AccountB	207
Accesso all'archivio tra account: azioni per l'utente del repository in AccountB	209
Eliminare un repository	215
Eliminare un CodeCommit repository (console)	215
Eliminare un repository locale	216
Eliminare un CodeCommit repository ()AWS CLI	216
Lavorare con i file	218
Sfoglia i file in un repository	219
Sfoglia un CodeCommit repository	220
Creare o aggiungere un file	221
Crea o carica un file (console)	221
Aggiungi un file ()AWS CLI	223
Aggiungi un file (Git)	224
Modifica il contenuto di un file	224
Modifica di un file (console)	225
Modifica o elimina un file ()AWS CLI	226
Modifica un file (Git)	228
Utilizzo delle richieste pull	229

Creazione di una richiesta di pull	233
Crea una pull request (console)	233
Crea una pull request ()AWS CLI	235
Creare una regola di approvazione	236
Crea una regola di approvazione per una pull request (console)	237
Crea una regola di approvazione per una pull request ()AWS CLI	239
Visualizzazione delle richieste pull	241
Visualizza le richieste pull (console)	241
Visualizza le richieste pull (AWS CLI)	242
Revisione di una richiesta di pull	247
Rivedi una pull request (console)	247
Rivedi le richieste pull ()AWS CLI	252
Aggiornamento di una richiesta di pull	257
Aggiorna una pull request (console)	258
Aggiorna le richieste pull ()AWS CLI	258
Modificare o eliminare una regola di approvazione	259
Modifica o elimina una regola di approvazione per una pull request (console)	260
Modifica o elimina una regola di approvazione per una pull request (AWS CLI)	261
Sostituisci le regole di approvazione su una pull request	263
Sostituisci le regole di approvazione (console)	263
Sostituisci le regole di approvazione ()AWS CLI	264
Unione di una richiesta di pull	265
Unisce una pull request (console)	266
Unisci una richiesta pull ()AWS CLI	269
Risolvi i conflitti in una pull request	275
Risolvi i conflitti in una pull request (console)	276
Risolvi i conflitti in una pull request ()AWS CLI	279
Chiusura di una richiesta di pull	287
Chiudi una pull request (console)	288
Chiudere una pull request (AWS CLI)	289
Utilizzo dei modelli di regola di approvazione	290
Creare un modello di regola di approvazione	292
Creare un modello di regola di approvazione (console)	292
Crea un modello di regola di approvazione ()AWS CLI	296
Associa un modello di regola di approvazione a un repository	298
Associare un modello di regola di approvazione (console)	298

Associa un modello di regola di approvazione (AWS CLI)	299
Gestire i modelli di regola di approvazione	300
Gestisci i modelli di regole di approvazione (console)	300
Gestisci i modelli di regole di approvazione (AWS CLI)	301
Dissociare un modello di regola di approvazione	305
Annulla l'associazione di un modello di regola di approvazione (console)	306
Dissocia un modello di regola di approvazione (AWS CLI)	306
Eliminare un modello di regola di approvazione	307
Eliminare un modello di regola di approvazione (console)	308
Elimina un modello di regola di approvazione (AWS CLI)	308
Lavorare con i commit	309
Crea un commit	310
Crea il primo commit per un repository utilizzando il AWS CLI	310
Crea un commit usando un client Git	312
Crea un commit usando il AWS CLI	315
Visualizzare i dettagli di commit	319
Sfoglia i commit in un repository	319
Visualizza i dettagli del commit (AWS CLI)	323
Visualizza i dettagli del commit (Git)	329
Confrontare i commit	331
Confronta un commit con il suo genitore	332
Confrontare due specificatori di commit	334
Commentare un commit	337
Visualizza i commenti su un commit in un repository	337
Aggiungi e rispondi ai commenti su un commit in un repository	338
Visualizza, aggiungi, aggiorna e rispondi ai commenti (AWS CLI)	343
Crea un tag Git	352
Usa Git per creare un tag	352
Visualizza i dettagli del tag	353
Visualizza i dettagli dei tag (console)	354
Visualizza i dettagli del tag Git (Git)	355
Eliminare un tag	357
Usa Git per eliminare un tag Git	357
Gestione dei rami	359
Creare un ramo	361
Crea un ramo (console)	361

Crea un ramo (Git)	362
Crea un ramo (AWS CLI)	363
Limita le spinte e le fusioni alle filiali	365
Configura una policy IAM per limitare i push e le unioni a una filiale	366
Applica la policy IAM a un gruppo o ruolo IAM	368
Prova la policy	368
Visualizza i dettagli della filiale	369
Visualizza i dettagli della filiale (console)	369
Visualizza i dettagli della filiale (Git)	370
Visualizza i dettagli della filiale (AWS CLI)	371
Confronta e unisci filiali	373
Confronta un ramo con il ramo predefinito	373
Confronta due rami specifici	373
Unisci due rami (AWS CLI)	374
Modificare le impostazioni della filiale	377
Cambia il ramo predefinito (console)	378
Cambia il ramo predefinito (AWS CLI)	378
Eliminare un ramo	379
Eliminare un ramo (console)	380
Elimina un ramo (AWS CLI)	380
Eliminare un ramo (Git)	381
Utilizzo delle preferenze dell'utente	383
Migrare a CodeCommit	384
Migrare un repository Git su AWS CodeCommit	384
Fase 0: Configurazione richiesta per l'accesso a CodeCommit	385
CodeCommit Fase 1: Creare un repository	391
Passaggio 2: clonare il repository e inviarlo al repository CodeCommit	393
Passaggio 3: Visualizza i file in CodeCommit	395
Fase 4: Condividere il repository CodeCommit	395
Esegui la migrazione dei contenuti in CodeCommit	398
Fase 0: Configurazione richiesta per l'accesso a CodeCommit	399
CodeCommit Fase 1: Creare un repository	405
Fase 2: Migrazione dei contenuti locali nel repository CodeCommit	406
Passaggio 3: Visualizza i file in CodeCommit	408
Passaggio 4: Condividi il repository CodeCommit	408
Esegui la migrazione di un repository in modo incrementale	411

Fase 0: determinare se effettuare la migrazione in modo incrementale	411
Fase 1: Installare i prerequisiti e aggiungere il CodeCommit repository come telecomando .	412
Passaggio 2: creare lo script da utilizzare per la migrazione incrementale	414
Passaggio 3: Esegui lo script ed esegui la migrazione incrementale a CodeCommit	415
Appendice: script di esempio incremental-repo-migration.py	416
Sicurezza	425
Protezione dei dati	425
AWS KMS e crittografia	426
Utilizzo di credenziali rotanti	429
Identity and Access Management	434
Destinatari	434
Autenticazione con identità	435
Gestione dell'accesso tramite policy	436
Autenticazione e controllo degli accessi	438
Come AWS CodeCommit funziona con IAM	504
Policy di CodeCommit basate sulle risorse	505
Autorizzazione basata su tag CodeCommit	505
CodeCommit Ruoli IAM	508
Esempi di policy basate su identità	509
Risoluzione dei problemi	512
Resilienza	515
Sicurezza dell'infrastruttura	515
Monitoraggio CodeCommit	517
Monitoraggio CodeCommit degli eventi	518
evento referenceCreated	519
evento referenceUpdated	520
evento referenceDeleted	520
unreferencedMergeCommitEvento creato	521
commentOnCommitEvento creato	522
commentOnCommitEvento aggiornato	523
commentOnPullRequestCreated evento	524
commentOnPullRequestUpdated evento	525
pullRequestCreated evento	525
pullRequestSourceBranchUpdated evento	527
pullRequestStatusEvento modificato	528
pullRequestMergeStatusUpdated evento	529

approvalRuleTemplateEvento creato	530
approvalRuleTemplateEvento aggiornato	530
approvalRuleTemplateEvento eliminato	531
approvalRuleTemplateAssociatedWithRepository evento	532
approvalRuleTemplateDisassociatedWithRepository evento	533
approvalRuleTemplateBatchAssociatedWithRepositories evento	534
approvalRuleTemplateBatchDisassociatedFromRepositories evento	535
pullRequestApprovalRuleCreated evento	535
pullRequestApprovalRuleDeleted evento	537
pullRequestApprovalRuleOverridden evento	538
pullRequestApprovalStateChanged evento	540
pullRequestApprovalRuleUpdated evento	542
Evento ReactionCreated	543
Evento ReactionUpdated	544
Registrazione delle chiamate AWS CodeCommit API con AWS CloudTrail	545
CodeCommit informazioni in CloudTrail	545
Comprendere CodeCommit le voci dei file di registro	546
AWS CloudFormation risorse	554
CodeCommit e CloudFormation modelli	554
Esempi di modello	555
CloudFormation, CodeCommit, e AWS Cloud Development Kit (AWS CDK)	556
Scopri di più su CloudFormation	557
Risoluzione dei problemi	558
Risoluzione dei problemi relativi alle credenziali Git (HTTPS)	558
Credenziali Git per AWS CodeCommit: Continuo a visualizzare una richiesta di credenziali quando mi connetto al mio CodeCommit repository dal terminale o dalla riga di comando ...	559
Credenziali Git per AWS CodeCommit: Ho impostato le credenziali Git, ma il mio sistema non le utilizza	559
Risoluzione dei problemi relativi a git-remote-codecommit	560
Vedo un errore: git: 'remote-codecommit' non è un comando git	560
Vedo un errore: fatale: impossibile trovare l'helper remoto per 'codecommit'	561
Errore di clonazione: non riesco a clonare un CodeCommit repository da un IDE	561
Errore push o pull: impossibile eseguire il push o il pull dei commit da un IDE in un repository CodeCommit	561
Errore di installazione: viene visualizzato un externally-managed-environment errore durante l'installazione git-remote-codecommit	562

Risoluzione dei problemi relativi alle connessioni SSH	562
Errore di accesso: la chiave pubblica viene caricata correttamente su IAM ma la connessione non riesce su sistemi Linux, macOS o Unix	563
Errore di accesso: la chiave pubblica è stata caricata correttamente su IAM e SSH è stato testato correttamente, ma la connessione non riesce sui sistemi Windows	564
Sfida di autenticazione: non è possibile stabilire l'autenticità dell'host durante la connessione a un repository CodeCommit	565
Errore IAM: 'Formato non valido' quando si tenta di aggiungere una chiave pubblica a IAM .	572
Devo accedere ai CodeCommit repository in più account Amazon Web Services con credenziali SSH	572
Git in Windows: l'emulatore Bash o la riga di comando si blocca durante il tentativo di connessione tramite SSH	573
Il formato della chiave pubblica richiede una specifica in alcune distribuzioni di Linux	574
Errore di accesso: chiave pubblica SSH negata durante la connessione a un repository CodeCommit	574
Risoluzione dei problemi relativi all'helper delle credenziali (HTTPS)	575
Ricevo un errore durante l'esecuzione del <code>git config</code> comando per configurare l'helper delle credenziali	575
Vedo un errore durante la lettura del nome utente durante il tentativo di clonare un repository	576
Ricevo un errore di comando non trovato in Windows durante l'utilizzo dell'assistente credenziali	576
Mi viene richiesto un nome utente quando mi connetto a un repository CodeCommit	577
Git per macOS: ho configurato correttamente l'helper delle credenziali, ma ora mi viene negato l'accesso al mio repository (403)	577
Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository (403)	581
Risoluzione dei problemi relativi ai client Git	583
Errore Git: Errore: RPC fallito; risultato = 56, codice HTTP = 200 fatale: l'estremità remota si è bloccata in modo imprevisto	584
Errore Git: troppi comandi di aggiornamento riferimenti	584
Errore Git: il push tramite HTTPS non funziona in alcune versioni di Git	584
Errore Git: "gnutls_handshake() failed" ("gnutls_handshake() non riuscito")	585
Errore Git: Git non riesce a trovare il CodeCommit repository o non ha il permesso di accedere al repository	585
Git in Windows: nessun metodo di autenticazione supportato disponibile (publickey)	585

Risoluzione degli errori di accesso	586
Errore di accesso: mi vengono richiesti un nome utente e una password quando mi connetto a un CodeCommit repository da Windows	586
Errore di accesso: chiave pubblica rifiutata durante la connessione a un repository CodeCommit	587
Errore di accesso: messaggio «Rate exceeded» o «429" durante la connessione a un repository CodeCommit	587
Risoluzione degli errori di configurazione	588
Errore di configurazione: impossibile configurare AWS CLI le credenziali su macOS	589
Risoluzione degli errori della console	589
Errore di accesso: accesso negato alla chiave di crittografia per un CodeCommit repository dalla console o AWS CLI	587
Errore di crittografia: il repository non può essere decrittografato	590
Errore della console: impossibile sfogliare il codice in un repository dalla console CodeCommit	590
Errore di visualizzazione: impossibile visualizzare un file o un confronto tra file	591
Risoluzione dei problemi relativi ai trigger	591
Errore del trigger: il trigger di un repository non viene eseguito quando previsto	591
Attiva il debug	592
CodeCommit riferimento	594
Regioni ed endpoint di connessione Git	594
Supportato per Regioni AWS CodeCommit	594
Endpoint di connessione Git	596
Impronte digitali del server per CodeCommit	603
Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia	610
Disponibilità	611
Crea endpoint VPC per CodeCommit	612
Creazione di una policy di endpoint VPC per CodeCommit.	613
Quote	614
Guida di riferimento alla riga di comando	623
Comandi Git di base	629
Variabili di configurazione	629
Archivi remoti	630
Commit	632
Rami	633
Tag	635

Cronologia dei documenti 636

 Aggiornamenti precedenti 648

AWS Glossario 655

..... dclvi

Che cos'è AWS CodeCommit?

AWS CodeCommit è un servizio di controllo delle versioni ospitato da Amazon Web Services che puoi utilizzare per archiviare e gestire in modo privato risorse (come documenti, codice sorgente e file binari) nel cloud. Per informazioni sui prezzi di CodeCommit, consulta la sezione [Prezzi](#).

Note

CodeCommit rientra nell'ambito di molti programmi di conformità. Per i dettagli AWS e gli sforzi di conformità, vedere [AWS Services In Scope by Compliance Program](#).

Questo è un servizio idoneo ai fini HIPAA. [Per ulteriori informazioni sull' AWS U.S. Health Insurance Portability and Accountability Act del 1996 \(HIPAA\) e sull'utilizzo AWS dei servizi per elaborare, archiviare e trasmettere informazioni sanitarie protette \(PHI\), vedere Panoramica HIPAA.](#)

Per informazioni su questo servizio e sullo standard ISO 27001, uno standard di gestione della sicurezza che specifica le migliori pratiche di gestione della sicurezza, consulta [Panoramica sulla ISO 27001](#).

Per ulteriori informazioni su questo servizio e Payment Card Industry Data Security Standard (PCI DSS), consulta la [panoramica di PCI DSS](#).

Per informazioni su questo servizio e lo standard di sicurezza del governo degli Stati Uniti Information Processing Standard (FIPS) Publication 140-2 che specifica i requisiti di sicurezza per moduli di crittografia per la protezione delle informazioni sensibili, consulta la [Panoramica del Federal Information Processing Standard \(FIPS 140-2\)](#) e [Endpoint di connessione Git](#).

Argomenti

- [Presentazione CodeCommit](#)
- [CodeCommit, Git e la scelta del AWS servizio giusto per le tue esigenze](#)
- [Come funziona? CodeCommit](#)
- [In cosa CodeCommit differisce il controllo delle versioni dei file in Amazon S3?](#)
- [Come posso iniziare? CodeCommit](#)
- [Dove posso saperne di più su Git?](#)

Presentazione CodeCommit

CodeCommit è un servizio di controllo del codice sorgente sicuro, altamente scalabile e gestito che ospita repository Git privati. CodeCommit elimina la necessità di gestire il proprio sistema di controllo del codice sorgente o di preoccuparsi della scalabilità della sua infrastruttura. È possibile utilizzarlo CodeCommit per archiviare qualsiasi cosa, dal codice ai file binari. Supporta le funzionalità standard di Git e si integra perfettamente con gli strumenti basati su Git esistenti.

Con CodeCommit, puoi:

- Approfita di un servizio completamente gestito ospitato da AWS. CodeCommit offre un'elevata disponibilità e durata del servizio ed elimina il sovraccarico amministrativo legato alla gestione di hardware e software propri. Non è necessario eseguire il provisioning dell'hardware né aumentarlo o ridurlo e non occorre installare, configurare e aggiornare software.
- Archivia il codice in modo sicuro. CodeCommit i repository sono crittografati sia a riposo che in transito.
- Lavora in modo collaborativo sul codice. CodeCommit I repository supportano le pull request, in cui gli utenti possono esaminare e commentare le reciproche modifiche al codice prima di unirle alle branch, le notifiche che inviano automaticamente e-mail agli utenti su richieste pull e commenti e altro ancora.
- Applicare facilmente la scalabilità ai progetti di controllo delle versioni. È possibile aumentare le risorse dei repository CodeCommit per soddisfare le esigenze di sviluppo. Il servizio è in grado di gestire repository con un numero elevato di file o rami, file di grandi dimensioni e lunghe cronologie di revisione.
- Archivia qualsiasi cosa, in qualsiasi momento. CodeCommit non ha limiti alla dimensione dei tuoi repository o ai tipi di file che puoi archiviare.
- Integrazione con altri servizi AWS e di terze parti. CodeCommit mantiene i tuoi repository vicini alle altre risorse di produzione nel AWS cloud, il che aiuta ad aumentare la velocità e la frequenza del ciclo di vita dello sviluppo. È integrato con IAM e può essere utilizzato con altri AWS servizi e in parallelo con altri repository. Per ulteriori informazioni, consulta [Integrazioni di prodotti e servizi con AWS CodeCommit](#).
- Eseguire facilmente la migrazione di file da altri repository remoti. Puoi migrare CodeCommit da qualsiasi repository basato su Git.
- Usa gli strumenti Git che già conosci. CodeCommit supporta i comandi Git oltre ai propri AWS CLI comandi e APIs.

CodeCommit, Git e la scelta del AWS servizio giusto per le tue esigenze

Essendo un servizio basato su Git, CodeCommit è adatto alla maggior parte delle esigenze di controllo delle versioni. Non sono previsti limiti arbitrari per dimensioni e tipo di file e per dimensioni del repository. Tuttavia, non vi sono limitazioni intrinseche a Git che possono compromettere le prestazioni di alcuni tipi di operazioni, soprattutto nel corso del tempo. È possibile evitare il potenziale peggioramento delle prestazioni del CodeCommit repository evitando di utilizzarlo in casi d'uso in cui altri AWS servizi sono più adatti all'attività. Inoltre puoi ottimizzare le prestazioni di Git per repository complessi. Ecco alcuni casi d'uso in cui Git CodeCommit, e quindi, potrebbe non essere la soluzione migliore per te o in cui potresti dover adottare misure aggiuntive per ottimizzare Git.

Caso d'uso	Description	Altri servizi da tenere in considerazione
File grandi modificati di frequente	Git utilizza la codifica delta per memorizzare differenze tra versioni di file. Ad esempio, se modifichi alcune parole in un documento, Git memorizza solo le parole modificate. Se disponi di file o oggetti di oltre 5 MB con molte modifiche, Git potrebbe dover ricostruire una grande catena di differenz e delta. Ciò può consumare una quantità crescente di risorse di elaborazione sia sul computer locale che in ingresso CodeCommit man mano che questi file crescono nel tempo.	Per creare versioni di file di grandi dimensioni, prendi in considerazione Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Using Versioning nella Guida per l'utente di Amazon Simple Storage Service.
Database	I repository Git aumentano di dimensioni nel tempo. Poiché la funzione Versioni	Per creare e utilizzare un database con prestazioni costanti indipendentemente

Caso d'uso	Description	Altri servizi da tenere in considerazione
	multiple tiene traccia di tutte le modifiche, qualsiasi modifica aumenterà le dimensioni del repository. In altre parole, quando esegui il commit dei dati, anche se elimini i dati di un commit, questi vengono aggiunti a un repository. Poiché c'è una maggiore quantità di dati da elaborare e trasmettere nel corso del tempo, Git rallenterà. Questo va a scapito soprattutto di un caso d'uso del database. Git non è stato progettato come database.	dalle dimensioni, prendi in considerazione Amazon DynamoDB. Per ulteriori informazioni, consulta la Amazon DynamoDB Getting Started Guide .
Audit trail	Di solito, gli audit trail vengono conservati per lunghi periodi di tempo e sono costantemente generati da processi di sistema a un ritmo molto frequente. Git è stato progettato per archiviare in modo sicuro il codice sorgente generato da gruppi di sviluppatori su un ciclo di sviluppo. I repository in rapido mutamento che memorizzano continuamente modifiche di sistema generate in modo programmatico offriranno prestazioni sempre peggiori.	Per archiviare gli audit trail, prendi in considerazione Amazon Simple Storage Service (Amazon S3). Per controllare AWS l'attività, a seconda del tuo caso d'uso, prendi in considerazione l'utilizzo AWS CloudTrail Idi Amazon o di Amazon CloudWatch . AWS Config

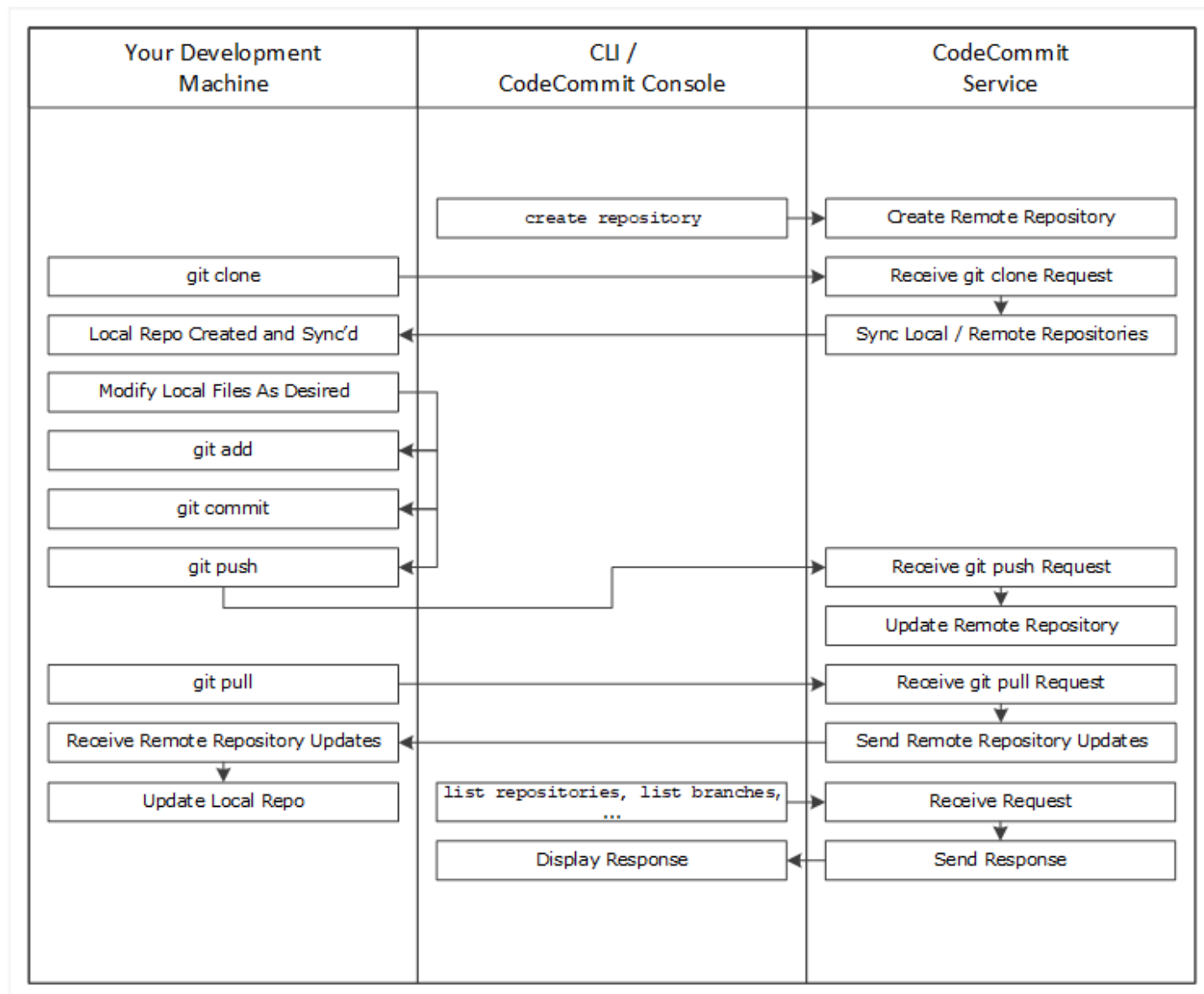
Caso d'uso	Description	Altri servizi da tenere in considerazione
Backup	Git è stato progettato per la versione del codice sorgente scritto dagli sviluppatori. È possibile inviare i commit a due repository remoti , incluso un CodeCommit repository, come strategia di backup. Tuttavia, Git non è stato progettato per gestire i backup del file system del computer, di dump di database o di contenuto simile. Questo tipo di utilizzo infatti potrebbe rallentare il sistema e aumentare l'intervallo di tempo richiesto per clonare ed eseguire il push di un repository.	Per informazioni sul backup nel AWS cloud, consulta Backup e ripristino .

Caso d'uso	Description	Altri servizi da tenere in considerazione
Molti rami o riferimenti	Quando un client Git esegue il push o il pull dei dati del repository, il server remoto deve inviare tutti i rami e i riferimenti, ad esempio i tag, anche se sei interessato solo a un singolo ramo. Se disponi di migliaia di rami e riferimenti, l'elaborazione e l'invio (negoiazione pacchetto) potrebbero richiedere tempo e rallentare apparentemente la risposta del repository. Più rami e tag sono presenti, più tempo potrebbe richiedere questo processo. Ti consigliamo di utilizzare CodeCommit, ma eliminare, rami e tag che non sono più necessari.	Per analizzare il numero di riferimenti in un CodeCommit repository e determinare quali potrebbero non essere necessari, puoi utilizzare uno dei seguenti comandi: - Emulatore Linux, macOS o Unix o Bash su Windows:<pre>git ls-remote wc -l</pre> - Powershell:<pre>git ls-remote Measure-Object -line</pre>

Come funziona? CodeCommit

CodeCommit è familiare agli utenti di repository basati su Git, ma anche quelli che non lo conoscono dovrebbero trovare la transizione relativamente semplice. CodeCommit fornisce una console per la facile creazione di repository e l'elenco di repository e filiali esistenti. In pochi semplici passaggi, gli utenti possono trovare informazioni su un repository e clonarle sul proprio computer, creando un repository locale in cui apportare le modifiche e quindi inviarle al repository. CodeCommit Gli utenti possono lavorare dalla riga di comando sui propri computer locali oppure usare un editor basato su un'interfaccia utente grafica.

La figura seguente mostra come utilizzare la macchina di sviluppo, la CodeCommit console AWS CLI o il CodeCommit servizio per creare e gestire gli archivi:



1. Usa la console AWS CLI o la CodeCommit console per creare un CodeCommit repository.
2. Dalla tua macchina di sviluppo, usa Git per eseguire `git clone`, specificando il nome del CodeCommit repository. Questo crea un repository locale che si connette al repository CodeCommit.
3. Usa il repository locale sulla tua macchina di sviluppo per modificare (aggiungere, modificare ed eliminare) i file, quindi esegui `git add` per archiviare i file modificati localmente. Esegui `git commit` per eseguire il commit dei file localmente, quindi esegui `git push` per inviare i file al CodeCommit repository.

4. Scaricare le modifiche apportate da altri utenti. Esegui git pull per sincronizzare i file nel repository con il CodeCommit repository locale. In questo modo sarai certo di utilizzare la versione più recente dei file.

Puoi usare la console AWS CLI o la CodeCommit console per tracciare e gestire i tuoi repository.

In cosa CodeCommit differisce il controllo delle versioni dei file in Amazon S3?

CodeCommit è ottimizzato per lo sviluppo di software in team. Gestisce batch di modifiche su più file, che possono avvenire in parallelo alle modifiche apportate da altri sviluppatori. Il controllo delle versioni di Amazon S3 supporta il ripristino di versioni precedenti dei file, ma non si concentra sulle funzionalità di tracciamento collaborativo dei file di cui hanno bisogno i team di sviluppo software.

Come posso iniziare? CodeCommit

Per iniziare con CodeCommit:

1. Seguire la procedura in [Configurazione](#) per preparare i computer di sviluppo.
2. Seguire le procedure indicate in uno o più dei tutorial disponibili in [Nozioni di base](#).
3. [Crea](#) progetti di controllo della versione CodeCommit o [migra](#) progetti di controllo della versione in CodeCommit.

Dove posso saperne di più su Git?

Se non hai ancora familiarità con questo sistema, [impara a usare Git](#). Ecco alcune utili risorse:

- [Pro Git](#), una versione online del manuale di Pro Git. Redatto da Scott Chacon. Pubblicato da Apress.
- [Git Immersion](#), un tour try-it-yourself guidato che ti guida attraverso i fondamenti dell'uso di Git. Pubblicata da Neo Innovation, Inc.
- [Git Reference](#), un riferimento rapido online utilizzabile anche come tutorial più approfondito su Git. Pubblicato dal team. GitHub
- [Git Cheat Sheet](#) con la sintassi dei principali comandi di Git. Pubblicato dal GitHub team.
- [Git Pocket Guide](#). Redatta da Richard E. Silverman. Pubblicata da O'Reilly Media, Inc.

Configurazione per AWS CodeCommit

È possibile accedere Console di gestione AWS e [caricare, aggiungere o modificare un file](#) in un repository direttamente dalla AWS CodeCommit console. Si tratta di un modo rapido per apportare modifiche. Tuttavia, se preferisci lavorare con più file, file in più rami e così via, ti consigliamo di configurare il computer locale per le operazioni con i repository. Il modo più semplice per eseguire la configurazione CodeCommit è configurare le credenziali HTTPS Git per AWS CodeCommit. Questo metodo di autenticazione HTTPS:

- Utilizza un nome utente e una password statici.
- Funziona con tutti i sistemi operativi supportati da CodeCommit.
- È anche compatibile con ambienti di sviluppo integrati (IDEs) e altri strumenti di sviluppo che supportano le credenziali Git.

Se non desideri o non puoi utilizzare le credenziali Git per motivi operativi, puoi utilizzare altri metodi. Ad esempio, se si accede agli CodeCommit archivi utilizzando accesso federato, credenziali temporanee o un provider di identità Web, non è possibile utilizzare le credenziali Git. Ti consigliamo di configurare il computer locale utilizzando il comando `git-remote-codecommit`. Leggi con attenzione queste opzioni, per stabilire quale metodo alternativo sia più adatto alle tue esigenze.

- [Configurazione tramite credenziali Git](#)
- [Configurazione con altri metodi](#)
- [Compatibilità per CodeCommit, Git e altri componenti](#)

Per informazioni sull'utilizzo CodeCommit di Amazon Virtual Private Cloud, consulta [Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia](#).

Visualizza e gestisci le tue credenziali

Puoi visualizzare e gestire le tue CodeCommit credenziali dalla AWS console tramite My Security Credentials.

 Note

Questa opzione non è disponibile per gli utenti che utilizzano accesso federato, credenziali temporanee o un provider di identità Web.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Selezionare il nome utente in alto a destra nella barra di navigazione e scegliere Security Credentials (Credenziali di sicurezza).
3. Scegli la scheda delle AWS CodeCommit credenziali.

Configurazione tramite credenziali Git

Con le connessioni HTTPS e le credenziali Git, generi un nome utente e una password statici in IAM. Potrai quindi utilizzare queste credenziali con Git e con qualsiasi strumento di terze parti che supporti l'autenticazione di nome utente e password Git. Questo metodo è supportato dalla maggior parte degli strumenti IDEs di sviluppo. È il metodo di connessione più semplice e facile da usare CodeCommit.

- [Per utenti HTTPS che utilizzano credenziali Git](#): Segui queste istruzioni per configurare le connessioni tra il computer locale e i CodeCommit repository utilizzando le credenziali Git.
- [Per connessioni da strumenti di sviluppo](#): Segui queste linee guida per configurare connessioni tra il tuo IDE o altri strumenti e CodeCommit repository di sviluppo utilizzando le credenziali Git. IDEs che supportano le credenziali Git includono (ma non sono limitati a) Visual Studio, Xcode e IntelliJ.

Configurazione con altri metodi

Puoi utilizzare il protocollo SSH anziché HTTPS per connetterti al tuo CodeCommit repository. Con le connessioni SSH, crei file di chiave pubblici e privati sulla tua macchina locale che Git e Git CodeCommit utilizzano per l'autenticazione SSH. Associate la chiave pubblica al vostro utente IAM. La chiave privata viene memorizzata nel computer locale. Poiché SSH richiede la creazione e la gestione manuale delle chiavi di file pubbliche e private, in genere le credenziali Git risultano più semplici da utilizzare con CodeCommit.

A differenza delle credenziali Git, la configurazione della connessione SSH varia in base al sistema operativo del computer locale.

- [Per gli utenti SSH che non utilizzano il AWS CLI](#): seguire le istruzioni abbreviate riportate di seguito se disponi già di una coppia di chiavi pubbliche-private e sono hai familiarità con le connessioni SSH sul computer locale.
- [Per connessioni SSH su Linux, macOS o Unix](#): Segui queste istruzioni per una step-by-step procedura dettagliata sulla creazione di una coppia di chiavi pubblica-privata e sulla configurazione delle connessioni su sistemi operativi Linux, macOS o Unix.
- [Per le connessioni SSH su Windows](#): Segui queste istruzioni per una step-by-step procedura dettagliata sulla creazione di key pair pubblica-privata e sulla configurazione delle connessioni sui sistemi operativi Windows.

Se ti connetti CodeCommit e AWS utilizzi l'accesso federato, un provider di identità o credenziali temporanee, o se non desideri configurare gli utenti IAM o le credenziali Git per gli utenti IAM, puoi configurare le connessioni ai CodeCommit repository in due modi:

- Installa e utilizza git-remote-codecommit (scelta consigliata).
- Installa e utilizza l'assistente credenziali incluso nell' AWS CLI.

Entrambi i metodi supportano l'accesso ai CodeCommit repository senza richiedere un utente IAM, il che significa che puoi connetterti ai repository utilizzando accesso federato e credenziali temporanee. L' git-remote-codecommit utilità è l'approccio consigliato. Estende Git ed è compatibile con una varietà di versioni Git e assistente credenziali. Tuttavia, non tutti IDEs supportano il formato URL clone utilizzato da git-remote-codecommit. Potrebbe essere necessario clonare manualmente i repository nel computer locale prima di poterli utilizzare nell'IDE.

- Segui le istruzioni in [Procedura di configurazione per le connessioni HTTPS ai AWS CodeCommit repository con git-remote-codecommit](#) per l'installazione e la configurazione git-remote-codecommit su Windows, Linux, macOS o Unix.

L'helper per le credenziali incluso in Git consente a AWS CLI Git di utilizzare HTTPS e una versione con firma crittografica delle credenziali utente IAM o del EC2 ruolo dell'istanza Amazon ogni volta che Git deve autenticarsi per interagire con i repository. AWS CodeCommit Alcuni sistemi operativi e alcune versioni Git dispongono dei propri assistenti credenziali, che tuttavia entrano in conflitto con quello incluso nella AWS CLI. Possono causare problemi di connettività per. CodeCommit

- [Per connessioni HTTPS su Linux, macOS o Unix con l'helper delle credenziali AWS CLI](#) : segui queste istruzioni per una step-by-step procedura dettagliata sull'installazione e la configurazione dell'helper delle credenziali su sistemi Linux, macOS o Unix.
- [Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI](#) : Segui queste istruzioni per una step-by-step procedura dettagliata sull'installazione e la configurazione dell'helper per le credenziali sui sistemi Windows.

Se ti connetti a un CodeCommit repository ospitato in un altro account Amazon Web Services, puoi configurare l'accesso e configurare le connessioni utilizzando ruoli, policy e l'helper per le credenziali incluso nel. AWS CLI

- [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#): Segui queste istruzioni per una step-by-step procedura dettagliata sulla configurazione dell'accesso tra account in un account Amazon Web Services per gli utenti di un gruppo IAM in un altro account Amazon Web Services.

Compatibilità per CodeCommit, Git e altri componenti

Quando lavori con CodeCommit, usi Git. Puoi anche utilizzare altri programmi. La tabella seguente fornisce le indicazioni più recenti per la compatibilità con le varie versioni. Come best practice, ti consigliamo di utilizzare le versioni più recenti di Git e altri software.

Informazioni sulla compatibilità delle versioni per AWS CodeCommit

Componente	Versione
Git	CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.
Curl	CodeCommit richiede curl 7.33 e versioni successive. Tuttavia, esiste un problema noto con HTTPS e l'aggiornamento curl 7.41.0. Per ulteriori informazioni, consulta Risoluzione dei problemi .

Componente	Versione
Python (solo git-remote-codecommit)	git-remote-codecommit richiede la versione 3 e successive.
Pip (solo git-remote-codecommit)	git-remote-codecommit richiede la versione 9.0.3 e successive.
AWS CLI (solo) git-remote-codecommit	Consigliamo una versione recente della AWS CLI versione 2 per tutti CodeCommit gli utenti. git-remote-codecommit richiede AWS CLI la versione 2 per supportare l' AWS SSO e le connessioni che richiedono credenziali temporanee, come gli utenti federati.

Configurazione per utenti HTTPS che utilizzano credenziali Git

Il modo più semplice per configurare le connessioni ai AWS CodeCommit repository consiste nel configurare le credenziali Git CodeCommit nella console IAM e quindi utilizzare tali credenziali per le connessioni HTTPS. Puoi utilizzare le stesse credenziali anche con qualsiasi strumento di terze parti o ambiente di sviluppo integrato (IDE) che supporti l'autenticazione HTTPS utilizzando un nome utente e una password statici. Per alcuni esempi, consulta [Per connessioni da strumenti di sviluppo](#).

Note

Se in precedenza hai configurato il tuo computer locale per l'utilizzo dell'helper delle credenziali CodeCommit, devi modificare il file.gitconfig per rimuovere le informazioni di supporto delle credenziali dal file prima di poter utilizzare le credenziali Git. Se il tuo computer locale esegue macOS, potresti dover cancellare le credenziali memorizzate nella cache da Keychain Access.

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per configurare un account Amazon Web Services, creare un utente IAM e configurare l'accesso a CodeCommit.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

 Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la politica da allegare, scegli Avanti: revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Se desideri utilizzare i AWS CLI comandi con CodeCommit, installa il AWS CLI. Ti consigliamo di creare un profilo per utilizzare AWS CLI with CodeCommit. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#) e [Utilizzo di profili denominati](#).

Fase 2: installazione di Git

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Passaggio 3: Creare credenziali Git per le connessioni HTTPS a CodeCommit

Dopo aver installato Git, crea le credenziali Git per il tuo utente IAM in IAM.

Per configurare le credenziali HTTPS Git per CodeCommit

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

Assicurati di accedere come utente IAM che creerà e utilizzerà le credenziali Git per le connessioni a CodeCommit.

2. Nella console IAM, nel pannello di navigazione, scegli Utenti e dall'elenco degli utenti, scegli il tuo utente IAM.

Note

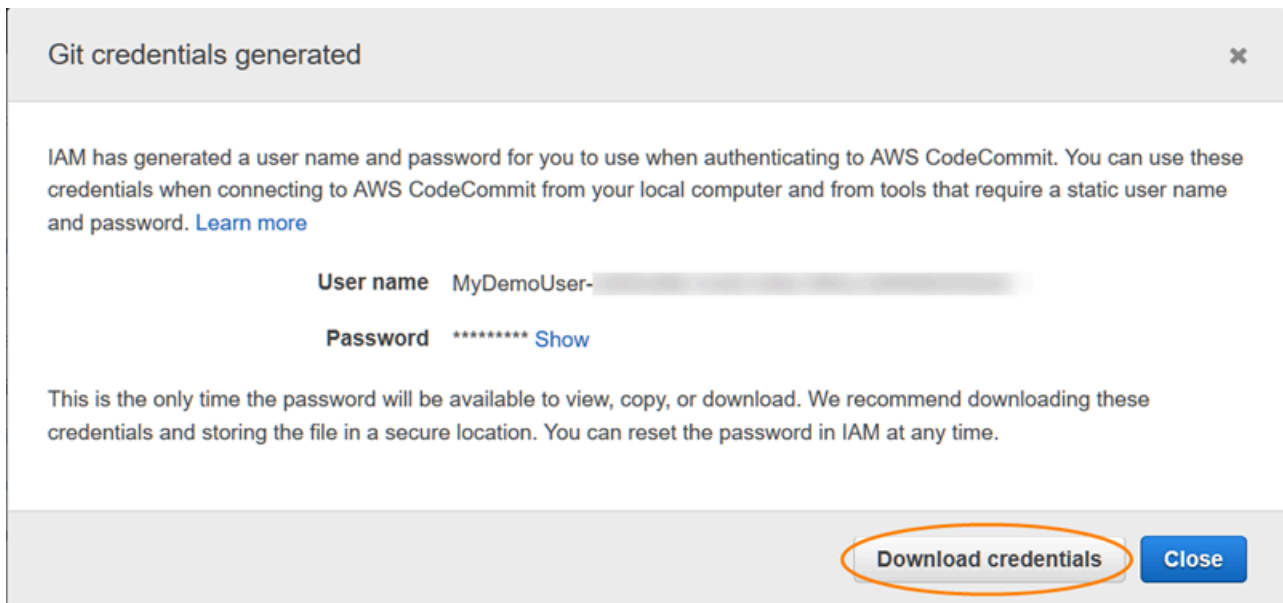
Puoi visualizzare e gestire direttamente le tue CodeCommit credenziali in My Security Credentials. Per ulteriori informazioni, consulta [Visualizza e gestisci le tue credenziali](#).

3. Nella pagina dei dettagli utente, scegli la scheda Credenziali di sicurezza e, in Credenziali Git HTTPS per AWS CodeCommit, scegli Genera.

**Note**

Non è possibile selezionare il nome utente e la password per le credenziali Git. Per ulteriori informazioni, consulta [Utilizzare credenziali Git e HTTPS con CodeCommit](#).

4. Copia il nome utente e la password che IAM ha generato per te, mostrando, copiando e incollando queste informazioni in un file sicuro sul tuo computer locale oppure scegliendo Scarica credenziali per scaricare queste informazioni come file.CSV. Queste informazioni sono necessarie per connettersi a CodeCommit.



Dopo aver salvato le credenziali, scegliere Close (Chiudi).

⚠ Important

Questa è l'unica occasione a disposizione per salvare il nome utente e la password. Se non le salvi, puoi copiare il nome utente dalla console IAM, ma non puoi cercare la password. Sarà quindi necessario reimpostare la password e salvarla.

Passaggio 4: Connect alla CodeCommit console e clona il repository

Se un amministratore ha già inviato il nome e i dettagli di connessione del repository CodeCommit, puoi ignorare questa fase e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.

- Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
- Copia l'URL HTTPS (GRC) se usi il comando `git-remote-codecommit` sul computer locale.
- Copia l'URL SSH se utilizzi una coppia di chiavi SSH pubblica/privata con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Apri un terminale, la riga di comando o la shell Git. Esegui il comando `git clone` con l'URL clone HTTPS copiato per clonare il repository. Ad esempio, per clonare un repository denominato su un repository locale denominato *MyDemoRepo my-demo-repo* nella regione Stati Uniti orientali (Ohio):

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

Alla prima connessione, verranno richiesti nome utente e password del repository. A seconda della configurazione del computer locale, questo prompt proviene da un sistema di gestione delle credenziali per il sistema operativo, da un'utilità di gestione delle credenziali per la versione di Git in uso (ad esempio, Git Credential Manager incluso in Git per Windows), dall'IDE o da Git stesso. Inserisci il nome utente e la password generati per le credenziali Git in IAM (quelle in cui hai creato [Passaggio 3: Creare credenziali Git per le connessioni HTTPS a CodeCommit](#)). In base al sistema operativo in uso e ad altri componenti software, queste informazioni potrebbero venire salvate in un archivio delle credenziali o in un'utilità di gestione delle credenziali. In tal caso, non dovresti ricevere una nuova richiesta a meno che non cambi la password, disattivi le credenziali Git o elimini le credenziali Git in IAM.

Se non disponi di un archivio delle credenziali o se nel computer locale non è configurata alcuna utilità di gestione delle credenziali, puoi installarne uno. Per ulteriori informazioni su Git e sulle relative modalità di gestione delle credenziali, consulta [Credential Storage](#) nella documentazione Git.

Per ulteriori informazioni, consulta [Connect al CodeCommit repository clonando il repository](#) e [Crea un commit](#).

Passaggi successivi

Hai completato i prerequisiti. Segui i passaggi indicati per iniziare a utilizzare [Iniziare con CodeCommit](#). CodeCommit

Per scoprire come creare e inviare il tuo primo commit, consulta [Crea un commit in AWS CodeCommit](#). Se non hai familiarità con Git, ti consigliamo di consultare le informazioni in [Dove posso saperne di più su Git?](#) e [Guida introduttiva a Git e AWS CodeCommit](#).

Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit

Se desideri connetterti CodeCommit utilizzando un account root, un accesso federato o credenziali temporanee, devi configurare l'accesso utilizzando `git-remote-codecommit`. Questa utilità fornisce un metodo semplice per inviare ed estrarre codice dai CodeCommit repository estendendo Git. È il metodo consigliato per supportare le connessioni effettuate con accesso federato, provider di identità e credenziali temporanee. Per assegnare autorizzazioni a un'identità federata, è necessario creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Create a role for a third-party identity provider \(federation\)](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center.

Puoi utilizzarlo anche `git-remote-codecommit` con un utente IAM. A differenza di altri metodi di connessione HTTPS, `git-remote-codecommit` non richiede l'impostazione delle credenziali Git per l'utente.

Note

Alcuni IDEs non supportano il formato URL clone utilizzato da `git-remote-codecommit`. Potrebbe essere necessario clonare manualmente i repository nel computer locale prima di

poterli utilizzare nell'IDE preferito. Per ulteriori informazioni, consulta [Risoluzione dei problemi git-remote-codecommit e AWS CodeCommit](#).

Queste procedure sono scritte presupponendo che tu disponga di un account Amazon Web Services, abbia creato almeno un repository e utilizzi un utente IAM con una policy gestita per la connessione ai CodeCommit repository. CodeCommit Per informazioni su come configurare l'accesso per gli utenti federati e altri tipi di rotazione delle credenziali, consulta [Connessione ai AWS CodeCommit repository con credenziali rotanti](#).

Argomenti

- [Fase 0: Installare i prerequisiti per git-remote-codecommit](#)
- [Fase 1: Configurazione iniziale per CodeCommit](#)
- [Fase 2: Installazione git-remote-codecommit](#)
- [Passaggio 3: Connect alla CodeCommit console e clona il repository](#)
- [Passaggi successivi](#)

Fase 0: Installare i prerequisiti per git-remote-codecommit

Prima di poter utilizzare git-remote-codecommit, è necessario installare alcuni prerequisiti nel computer locale. Ciò include:

- Una versione supportata di Python. Per ulteriori informazioni sulle versioni supportate di Python, vedere. [git-remote-codecommit](#)

Per ulteriori informazioni sulla configurazione e l'utilizzo git-remote-codecommit, vedere [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

- Git

Note

Quando installi Python su Windows, assicurati di scegliere l'opzione per aggiungere Python al percorso.

git-remote-codecommit richiede pip versione 9.0.3 o successiva. Per controllare la versione di pip, apri un terminale o una riga di comando ed esegui il comando seguente:

```
pip --version
```

Puoi eseguire i due comandi seguenti per aggiornare pip alla versione più recente:

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

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per creare un utente IAM, configurarlo con le politiche appropriate, ottenere una chiave di accesso e una chiave segreta e installare e configurare AWS CLI.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

 Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che all'utente non siano associate policy che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la policy da allegare, scegli Avanti: Revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Per installare e configurare il AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con `git-remote-codecommit`

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

 Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o AWS CLI aggiorna l'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando. Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il `configure` comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS dove si trova il repository, ad esempio. `us-east-2` Quando viene richiesto il formato di output predefinito, specificare `json`. Ad esempio, se stai configurando un profilo per un utente 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
```

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)

- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Fase 2: Installazione git-remote-codecommit

Segui queste fasi per installare git-remote-codecommit.

Per installare git-remote-codecommit

1. Dal terminale o dalla riga di comando, eseguire il comando seguente:

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

Note

A seconda del sistema operativo e della configurazione, potrebbe essere necessario eseguire questo comando con autorizzazioni elevate, come sudo, o utilizzare il parametro --user per l'installazione in una directory che non richiede privilegi speciali, come l'account utente corrente. Ad esempio, su un computer che esegue Linux, macOS o Unix:

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

Su un computer con sistema operativo Windows:

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

2. Monitora il processo di installazione finché non viene visualizzato un messaggio di successo.

Passaggio 3: Connect alla CodeCommit console e clona il repository

Se un amministratore ti ha già inviato l'URL del clone da utilizzare git-remote-codecommit per il CodeCommit repository, puoi saltare la connessione alla console e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.
 - Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
 - Copia l'URL HTTPS (GRC) se usi il comando git-remote-codecommit sul computer locale.
 - Copia l'URL SSH se utilizzi una coppia di chiavi SSH pubblica/privata con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in cui hai effettuato l'AWS accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Al terminale o al prompt dei comandi, clonare il repository con il comando git clone. Usa l' git-remote-codecommitURL HTTPS che hai copiato e il nome del AWS CLI profilo, se hai creato un profilo denominato. Se non si specifica un profilo, il comando utilizza il profilo predefinito. Il repository locale viene creato in una sottodirectory della directory in cui viene eseguito il comando. Ad esempio, per clonare il repository *MyDemoRepo* nel repository locale *my-demo-repo*:

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

Per clonare lo stesso repository utilizzando un profilo denominato: *CodeCommitProfile*


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

Per clonare un repository in un repository Regione AWS diverso da quello configurato nel tuo profilo, includi il nome. Regione AWS Per esempio:

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

Passaggi successivi

Hai completato i prerequisiti. Segui i passaggi indicati [Iniziare con CodeCommit](#) per iniziare a utilizzare. CodeCommit

Per scoprire come creare e inviare il tuo primo commit, consulta [Crea un commit in AWS CodeCommit](#). Se non hai familiarità con Git, ti consigliamo di consultare le informazioni in [Dove posso saperne di più su Git?](#) e [Guida introduttiva a Git e AWS CodeCommit](#).

Configura le connessioni dagli strumenti di sviluppo utilizzando le credenziali Git

Dopo aver configurato le credenziali Git per AWS CodeCommit nella console IAM, puoi utilizzare tali credenziali con qualsiasi strumento di sviluppo che supporti le credenziali Git. Ad esempio, puoi configurare l'accesso al tuo CodeCommit repository in Visual Studio AWS Cloud9, Xcode, IntelliJ o qualsiasi ambiente di sviluppo integrato (IDE) che integri le credenziali Git. Dopo avere configurato l'accesso, puoi modificare il codice, eseguire il commit delle modifiche e il push direttamente dall'IDE o da un altro strumento di sviluppo.

Note

Se si accede ai CodeCommit repository utilizzando accesso federato, credenziali temporanee o un provider di identità Web, non è possibile utilizzare le credenziali Git. Ti consigliamo di configurare il computer locale utilizzando il comando `git-remote-codecommit`. Tuttavia, non tutti IDEs sono completamente compatibili con gli helper remoti Git come `git-remote-codecommit`. Se riscontri problemi, consulta [Risoluzione dei problemi git-remote-codecommit e AWS CodeCommit](#).

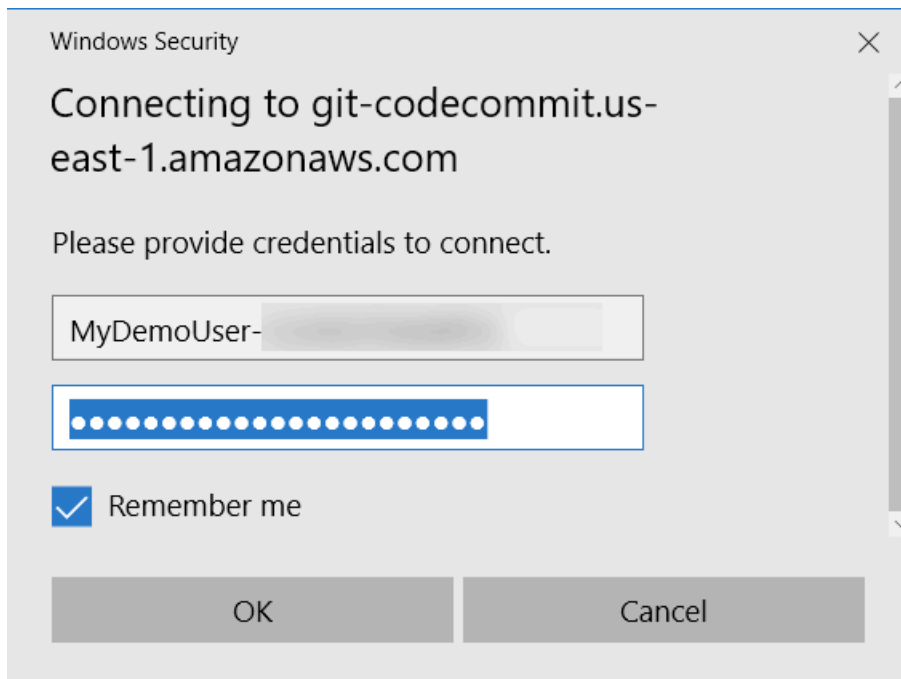
Argomenti

- [Integrazione AWS Cloud9 con AWS CodeCommit](#)
- [Integra Visual Studio con AWS CodeCommit](#)

Quando l'IDE o lo strumento di sviluppo richiedono il nome utente e la password utilizzati per connettersi al CodeCommit repository, fornisci le credenziali Git per il nome utente e la password che hai creato in IAM.

Per ulteriori informazioni sugli Regioni AWS endpoint per cui, consulta. CodeCommit [Regioni ed endpoint di connessione Git](#)

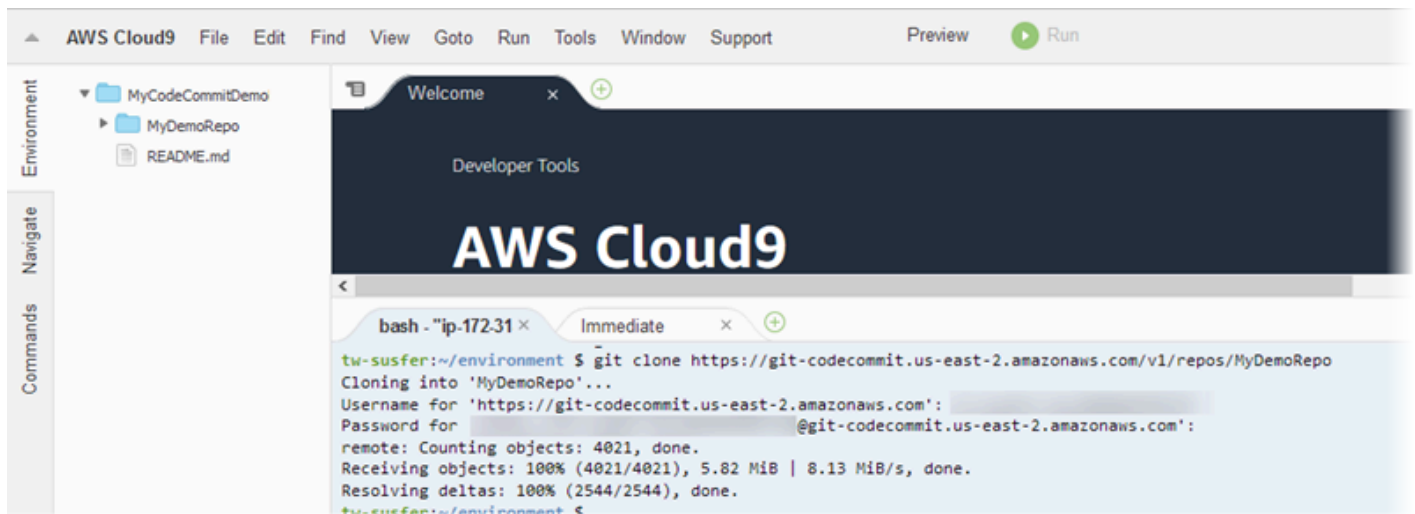
Potrebbe inoltre venire visualizzata una richiesta di memorizzazione del nome utente e della password da parte del sistema operativo. Ad esempio, in Windows, potresti fornire le credenziali Git come segue:



Per informazioni sulla configurazione delle credenziali Git per un determinato programma software o un altro strumento di sviluppo, consulta la documentazione di prodotto.

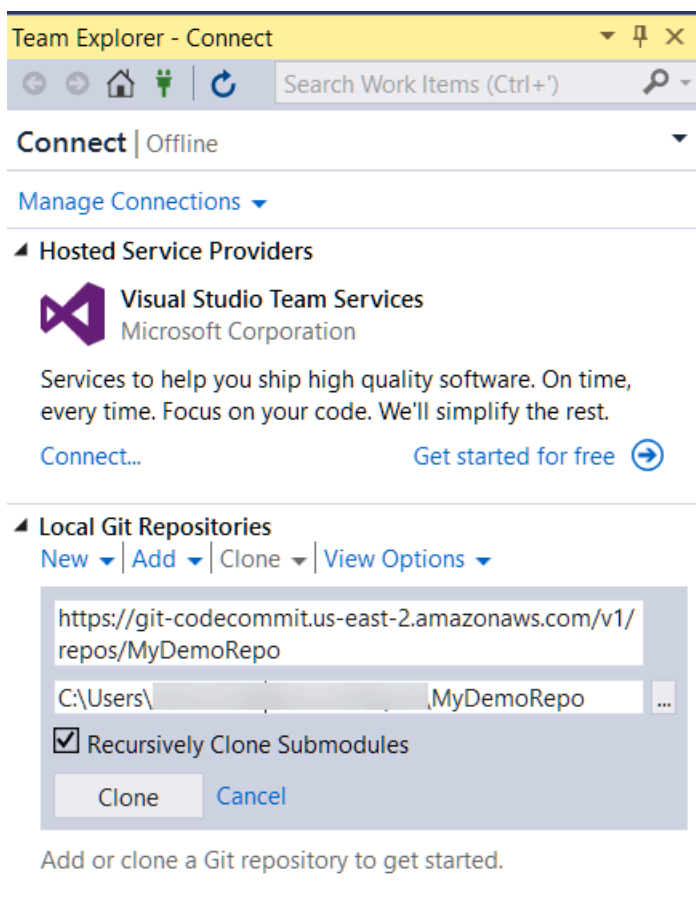
Di seguito non è riportato un elenco completo di IDEs. I collegamenti vengono forniti esclusivamente per aiutarti a saperne di più su questi strumenti. AWS non è responsabile per il contenuto di nessuno di questi argomenti.

- [AWS Cloud9](#)



- [Visual Studio](#)

In alternativa, installa il [AWS Toolkit for Visual Studio](#). Per ulteriori informazioni, consulta [Integra Visual Studio con AWS CodeCommit](#).



- [XCode](#)

Integrazione AWS Cloud9 con AWS CodeCommit

Puoi usare AWS Cloud9 per apportare modifiche al codice in un CodeCommit repository. AWS Cloud9 contiene una raccolta di strumenti che è possibile utilizzare per scrivere codice e creare, eseguire, testare, eseguire il debug e rilasciare software. Puoi clonare repository esistenti, creare repository, eseguire il commit e inviare modifiche al codice in un repository e altro ancora, tutto dal tuo ambiente di sviluppo. AWS Cloud9 EC2 L'ambiente di AWS Cloud9 EC2 sviluppo è generalmente preconfigurato con un EC2 ruolo Amazon e Git, quindi nella maggior parte dei casi puoi eseguire alcuni semplici comandi e iniziare a interagire con il tuo repository. AWS CLI

Per utilizzarlo AWS Cloud9 con CodeCommit, è necessario quanto segue:

- Un ambiente di AWS Cloud9 EC2 sviluppo in esecuzione su Amazon Linux.
- L' AWS Cloud9 IDE si apre in un browser Web.
- Un utente IAM a cui sono state applicate una delle policy CodeCommit AWS Cloud9 gestite e una delle policy gestite.

Per ulteriori informazioni, vedi [AWS politiche gestite per CodeCommit](#) e [Comprendere e ottenere le credenziali di sicurezza](#).

Note

Questo argomento descrive la configurazione dell'integrazione AWS Cloud9 con CodeCommit e con accesso generale da Internet. È possibile configurare l'accesso a CodeCommit e AWS Cloud9 in un ambiente isolato, ma ciò richiede passaggi aggiuntivi. Per ulteriori informazioni, consultare:

- [Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia](#)
- [Accesso a istanze Amazon EC2 senza ingresso con AWS Systems Manager](#)
- [Lavorare con gli ambienti condivisi](#)
- [Condividi il tuo VPC con altri account](#)
- [Post sul blog: Isolamento dell'accesso alla rete ai tuoi ambienti AWS Cloud9](#)

Argomenti

- [Fase 1: Creare un ambiente di AWS Cloud9 sviluppo](#)

- [Passaggio 2: configura l'helper AWS CLI delle credenziali nel tuo ambiente di sviluppo AWS Cloud9 EC2](#)
- [Fase 3: clonare un repository nel proprio ambiente di sviluppo CodeCommit AWS Cloud9 EC2](#)
- [Passaggi successivi](#)

Fase 1: Creare un ambiente di AWS Cloud9 sviluppo

AWS Cloud9 ospita il tuo ambiente di sviluppo su un' EC2 istanza Amazon. Questo è il modo più semplice per l'integrazione, perché puoi utilizzare le credenziali temporanee AWS gestite per consentire all'istanza di connettersi al tuo CodeCommit repository. Se invece desideri utilizzare il tuo server, consulta la Guida per l'[AWS Cloud9 utente](#).

Per creare un AWS Cloud9 ambiente

1. Accedi AWS come utente IAM che hai configurato e apri la AWS Cloud9 console.
2. Nella AWS Cloud9 console, scegli Crea ambiente.
3. In Step 1: Name environment (Fase 1: Nome ambiente), immetti un nome e una descrizione facoltativa per l'ambiente, quindi scegli Next step (Fase successiva).
4. In Step 2: Configure Settings (Fase 2: Configura impostazioni), configura l'ambiente come segue:
 - In Tipo di ambiente, scegli Crea una nuova istanza per environment (EC2).
 - In Instance type (Tipo di istanza), seleziona il tipo di istanza del tuo ambiente di sviluppo. Ad esempio, se stai solo esplorando il servizio, potresti scegliere l'istanza t2.micro predefinita. Se hai intenzione di utilizzare questo ambiente per lavori di sviluppo, scegli un tipo di istanza di dimensioni maggiori.
 - Accetta le altre impostazioni predefinite a meno che tu non abbia motivi per scegliere diversamente (ad esempio, la tua organizzazione utilizza un VPC specifico o il tuo account Amazon Web Services non ne ha VPCs configurato alcuno), quindi scegli Passaggio successivo.
5. In Step 3: Review (Fase 3: Revisione), rivedi le impostazioni. Se devi apportare modifiche, scegli Previous step (Fase precedente). In caso contrario, seleziona Create environment (Crea ambiente).

La creazione di un ambiente e la prima connessione allo stesso richiedono alcuni minuti. Se il tempo richiesto è superiore a quello previsto, consulta la sezione [Risoluzione dei problemi](#) nella Guida per l'utente di AWS Cloud9 .

6. Dopo aver effettuato la connessione all'ambiente, controlla se Git è già installato con una versione supportata eseguendo il comando `git --version` nella finestra del terminale.

Se Git non è installato o se non è una versione supportata, installa una versione supportata. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git. Per installare Git, consigliamo siti web come [Git Downloads](#).

 Tip

A seconda del sistema operativo dell'ambiente, potresti essere in grado di utilizzare il comando `yum` con l'opzione `sudo` per installare gli aggiornamenti, fra cui Git. Ad esempio, una sequenza di comandi a livello amministrativo potrebbe assomigliare ai seguenti tre comandi:

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

7. Configura un nome utente e un indirizzo e-mail da assegnare ai commit Git eseguendo il comando `git config`. Per esempio:

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

Passaggio 2: configura l'helper AWS CLI delle credenziali nel tuo ambiente di sviluppo AWS Cloud9 EC2

Dopo aver creato un AWS Cloud9 ambiente, puoi configurare l'helper per le AWS CLI credenziali per gestire le credenziali per le connessioni al tuo repository. CodeCommit L'ambiente di AWS Cloud9 sviluppo include credenziali temporanee AWS gestite associate al tuo utente IAM. Queste credenziali vengono utilizzate con l'helper delle AWS CLI credenziali.

1. Apri la finestra del terminale ed esegui il seguente comando per verificare di avere installato AWS CLI :

```
aws --version
```

In caso di esito positivo, questo comando restituisce la versione correntemente installata di AWS CLI. [Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedere Installazione di. AWS Command Line Interface](#)

2. Sul terminale, esegui i seguenti comandi per configurare l'helper delle AWS CLI credenziali per le connessioni HTTPS:

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

Tip

L'helper delle credenziali utilizza il ruolo predefinito dell' EC2 istanza Amazon per il tuo ambiente di sviluppo. Se intendi utilizzare l'ambiente di sviluppo per connetterti a repository che non sono ospitati in CodeCommit, configura le connessioni SSH a tali repository o configura un `.gitconfig` file locale per utilizzare un sistema di gestione delle credenziali alternativo durante la connessione a quegli altri repository. Per ulteriori informazioni, consulta [Git Tools - Credential Storage](#) sul sito Web Git.

Fase 3: clonare un repository nel proprio ambiente di sviluppo CodeCommit AWS Cloud9 EC2

Dopo aver configurato l'helper delle AWS CLI credenziali, puoi clonare il tuo repository su di esso. CodeCommit A questo punto puoi iniziare a lavorare con il codice.

1. Nel terminale, esegui il comando `git clone`, specificando l'URL clone HTTPS del repository da clonare. Ad esempio, se desideri clonare un repository denominato `MyDemoRepo` nella regione Stati Uniti orientali (Ohio), devi inserire:

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

 Tip

Puoi trovare l'URL del clone per il tuo repository nella CodeCommit console selezionando Clone URL.

2. Al termine della clonazione, nel riquadro di navigazione, espandi la cartella del repository nel riquadro di navigazione laterale e scegli il file da aprire per la modifica. In alternativa, scegli File, quindi New File (Nuovo file) per creare un file.
3. Una volta completata la modifica o la creazione di file, nella finestra del terminale, modifica le directory del repository clonato, quindi esegui il commit e il push delle modifiche. Ad esempio, se hai aggiunto un nuovo file denominato: *MyFile.py*

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

Passaggi successivi

Per ulteriori informazioni, consultate la [Guida per AWS Cloud9 l'utente](#) e [CodeCommit un esempio](#) di AWS Cloud9. Per ulteriori informazioni sull'utilizzo di Git con CodeCommit, consulta [Guida introduttiva a Git e AWS CodeCommit](#).

Integra Visual Studio con AWS CodeCommit

Puoi usare Visual Studio per apportare modifiche al codice in un CodeCommit repository. AWS Toolkit for Visual Studio Ora include funzionalità che semplificano e CodeCommit semplificano l'utilizzo quando si lavora in Visual Studio. L'integrazione di Toolkit for Visual Studio è progettato per funzionare con credenziali Git e un utente IAM. Puoi clonare repository esistenti, creare repository, eseguire il commit e il push delle modifiche al codice in un repository e molto altro.

 Important

Toolkit for Visual Studio è disponibile per l'installazione solo su sistemi operativi Windows. Se stai cercando informazioni sull'utilizzo di Visual Studio Code, consulta [AWS Toolkit for Visual Studio Code](#).

Se hai già usato Toolkit for Visual Studio, probabilmente conosci già la AWS configurazione di profili di credenziali che contengono una chiave di accesso e una chiave segreta. I profili di credenziali vengono utilizzati nel Toolkit for Visual Studio per abilitare le chiamate AWS al APIs servizio (ad esempio, ad Amazon S3 per elencare i bucket o CodeCommit per elencare gli archivi). Per estrarre e inviare codice a un CodeCommit repository, sono necessarie anche le credenziali Git. Se non disponi di credenziali Git, Toolkit for Visual Studio può generare e applicare tali credenziali per te. Ciò si traduce in un notevole risparmio di tempo.

Per usare Visual Studio con CodeCommit, è necessario quanto segue:

- Un utente IAM con un set di credenziali valido (una chiave di accesso e una chiave segreta) configurato per questo. Questo utente IAM dovrebbe inoltre avere:

Una delle politiche CodeCommit gestite e la IAMSelfManageServiceSpecificCredentials politica gestita applicata ad essa.

O

Se l'utente IAM ha già configurato le credenziali Git, una delle policy CodeCommit gestite o autorizzazioni equivalenti.

Per ulteriori informazioni, vedi [AWS politiche gestite per CodeCommit](#) e [Comprendere e ottenere le credenziali di sicurezza](#).

- È AWS Toolkit for Visual Studio installato sul computer in cui è stato installato Visual Studio. Per ulteriori informazioni, consulta [Configurazione di AWS Toolkit for Visual Studio](#).

Per ulteriori informazioni sull'utilizzo AWS Toolkit for Visual Studio con CodeCommit, consulta [Using AWS CodeCommit with Visual Studio Team Explorer](#) nella Toolkit for Visual Studio User Guide.

Configurazione per utenti SSH che non utilizzano AWS CLI

Se desideri utilizzare connessioni SSH per il tuo repository, puoi connetterti a AWS CodeCommit senza installare il. AWS CLI AWS CLI Include comandi utili per l'utilizzo e la gestione degli CodeCommit archivi, ma non necessari per la configurazione iniziale.

Questo argomento presuppone quanto segue:

- Hai configurato un utente IAM con le policy o le autorizzazioni richieste CodeCommit e la policy IAMUserSSHKeysgestita o autorizzazioni equivalenti richieste per il caricamento delle chiavi. Per ulteriori informazioni, consulta [Utilizzo di policy basate sull'identità \(IAM Policies\) per CodeCommit](#).
- Essere in grado di creare o avere già creato una coppia di chiavi pubbliche/private. Consigliamo di utilizzare una passphrase sicura per la chiave SSH.
- Avere acquisito familiarità con SSH, il client Git e i relativi file di configurazione.
- Se utilizzi Windows, l'installazione di una utilità della riga di comando, come Git Bash, che emula la shell bash.

Per ulteriori informazioni, consulta le istruzioni in [Per connessioni SSH su Linux, macOS o Unix](#) o [Per le connessioni SSH su Windows](#).

Argomenti

- [Passaggio 1: associa la tua chiave pubblica al tuo utente IAM](#)
- [Passaggio 2: aggiungi CodeCommit alla tua configurazione SSH](#)
- [Fasi successive](#)

Passaggio 1: associa la tua chiave pubblica al tuo utente IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nella console IAM, nel pannello di navigazione, scegli Utenti e dall'elenco degli utenti scegli il tuo utente IAM.
3. Nella scheda Security Credentials (Credenziali di sicurezza), scegli Upload SSH public key (Carica chiave pubblica SSH).
4. Incolla il contenuto della chiave pubblica SSH nel campo, quindi scegli Upload SSH key (Carica chiave SSH).

Tip

La coppia di chiavi pubbliche/private deve essere SSH-2 RSA, in formato OpenSSH, e includere 2048 bit. L'aspetto della chiave è simile al seguente:

```
ssh-rsa EXAMPLE-  
AfICcQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJB
```

```
gnVBAGtAldBMRaWdgYdVQqHEwdTZWF0dGx1MQ8wDQYdVQqKEwZBbWF6b24xFDASBgNVBAstC01BTSBDb2
5zb2x1MRIwEAYdVQqDEw1UZxN0Q21sYWxhZAdBgqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhc
NMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxwJCBiDELMAkGA1UEBhMCMVVMxzcAJBgNVBAgtAldBMRaW
dgYdVQqHEwdTZWF0dGx1MQ8wDQYdVQqKEwZBbWF6b24xFDAS=EXAMPLE  user-
name@ip-192-0-2-137
```

IAM accetta chiavi pubbliche solo nel formato OpenSSH. Se fornisci la chiave pubblica in un altro formato, viene visualizzato un messaggio di errore che indica che il formato della chiave non è valido.

5. Copia l'ID della chiave SSH (ad esempio `APKAEIBAERJR2EXAMPLE`) e chiudi la console.

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

Passaggio 2: aggiungi CodeCommit alla tua configurazione SSH

1. Al terminale (Linux, macOS o Unix) o all'emulatore bash (Windows), modifica il file di configurazione SSH digitando: `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

Se disponi di più configurazioni SSH, assicurati di includere le righe vuote prima e dopo il contenuto. Salva il file premendo contemporaneamente i tasti **Ctrl** e **d**.

2. Esegui il seguente comando per verificare la configurazione SSH:

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

Quando richiesto, immetti la passphrase del file della chiave SSH. Se tutto è configurato correttamente, viene visualizzato il seguente messaggio:

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

Fasi successive

Hai completato i prerequisiti. Segui i passaggi indicati per iniziare a utilizzare. [Iniziare con CodeCommit](#) CodeCommit

Per effettuare la connessione a un repository, segui le fasi descritte in [Connessione a un repository](#). Per creare un repository, segui le fasi descritte in [Creazione di un repository](#).

Passaggi di configurazione per le connessioni SSH ai AWS CodeCommit repository su Linux, macOS o Unix

Prima di poterti connettere CodeCommit per la prima volta, devi completare alcuni passaggi iniziali di configurazione. Dopo aver configurato il computer e il AWS profilo, è possibile connettersi a un CodeCommit repository e clonarlo sul computer (operazione nota anche come creazione di un repository locale). Se non hai mai utilizzato Git, ti consigliamo di consultare le informazioni in [Dove posso saperne di più su Git?](#).

Argomenti

- [Fase 1: Configurazione iniziale per CodeCommit](#)
- [Fase 2: installazione di Git](#)
- [Passaggio 3: configurare le credenziali su Linux, macOS o Unix](#)
- [Passaggio 4: Connect alla CodeCommit console e clona il repository](#)
- [Fasi successive](#)

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per configurare un account Amazon Web Services, creare un utente IAM e configurare l'accesso a CodeCommit.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

 Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la policy da allegare, scegli Avanti: Revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

 Note

Se desideri utilizzare i AWS CLI comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Fase 2: installazione di Git

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

 Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Passaggio 3: configurare le credenziali su Linux, macOS o Unix

SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit

Per configurare le chiavi pubbliche e private per Git e CodeCommit

1. Dal terminale del computer locale, esegui il comando `ssh-keygen` e segui le indicazioni per salvare il file nella directory `.ssh` del tuo profilo.

 Note

Assicurati di chiedere all'amministratore di sistema dove archiviare i file delle chiavi e quale schema di denominazione dei file devi utilizzare.

Esempio:

```
$ 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 . . . |
|      .         |
|               |
|               |
|               |
+-----+

```

Questo genera:

- Il *codecommit_rsa* file, che è il file con chiave privata.
- Il *codecommit_rsa.pub* file, che è il file con chiave pubblica.

 Tip

Per impostazione predefinita, ssh-keygen genera una chiave a 2048 bit. È possibile utilizzare i parametri -t e -b per specificare il tipo e la lunghezza della chiave. Se desideri

una chiave a 4096 bit nel formato rsa, devi specificarla eseguendo il comando con i seguenti parametri:

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

[Per ulteriori informazioni sui formati e le lunghezze richieste per le chiavi SSH, consulta Using IAM with. CodeCommit](#)

2. Esegui il comando seguente per visualizzare il valore del file della chiave pubblica (`codecommit_rsa.pub`):

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

Copia questo valore. Saranno simili a quanto segue:

```
ssh-rsa EXAMPLE-AfICCQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ2sYWMxHZAAdBgqhkiG9w0BCQEWEG5vb25lQGFtYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE user-name@ip-192-0-2-137
```

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

Note

Puoi visualizzare e gestire direttamente le tue CodeCommit credenziali in Le mie credenziali di sicurezza. Per ulteriori informazioni, consulta [Visualizza e gestisci le tue credenziali](#).

4. Nella console IAM, nel riquadro di navigazione, scegli Utenti e dall'elenco degli utenti scegli il tuo utente IAM.
5. Nella pagina dei dettagli dell'utente, seleziona la scheda Security Credentials (Credenziali di sicurezza), quindi Upload SSH public key (Carica chiave pubblica SSH).
6. Incolla il contenuto della chiave pubblica SSH nel campo, quindi scegli Upload SSH public key (Carica chiave pubblica SSH).
7. Copia o salva le informazioni in SSH Key ID (ad esempio, `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
APKAEIBAERJR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make Inactive Show SSH Key Delete

 Note

Se hai IDs caricato più di una chiave SSH, le chiavi sono elencate alfabeticamente per ID chiave, non per data di caricamento. Assicurati di aver copiato l'ID della chiave associato alla data di caricamento corretta.

- Sul computer locale, usa un editor di testo per creare un file di configurazione nella directory `~/.ssh`, quindi aggiungi le seguenti righe al file, dove il valore per *User* è l'ID della chiave SSH che hai copiato in precedenza:

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

 Note

Se hai dato al file della chiave privata un nome diverso da *codecommit_rsa*, assicurati di usarlo qui.

Puoi configurare l'accesso SSH ai repository in più account Amazon Web Services. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit](#)

Salva e attribuisce un nome al file config.

- Dal terminale, esegui il seguente comando per modificare le autorizzazioni per il file config:

```
chmod 600 config
```

- Esegui il seguente comando per verificare la configurazione SSH:

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

Ti viene chiesto di confermare la connessione perché non `git-codecommit.us-east-2.amazonaws.com` è ancora inclusa nel tuo file `hosts` noto. L'impronta digitale del CodeCommit server viene visualizzata come parte della verifica (a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e per MD5 o 31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UES56fG6ZIZQ per SHA256).

Note

CodeCommit le impronte digitali del server sono uniche per tutti. Regione AWS Per visualizzare le impronte digitali del server per un Regione AWS, vedere. [Impronte digitali del server per CodeCommit](#)

Dopo avere verificato la connessione, dovrebbe venire visualizzato un messaggio di conferma dell'avvenuta aggiunta del server ai file `hosts` noti e un messaggio dell'avvenuta connessione. Se non visualizzi un messaggio di successo, verifica di aver salvato il `config` file nella directory `~/.ssh` dell'utente IAM a cui hai configurato l'accesso CodeCommit e di aver specificato il file di chiave privata corretto.

Per informazioni su come risolvere i problemi, esegui il comando con il parametro. `ssh -v`
Esempio:

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

Per informazioni sulla risoluzione dei problemi di connessione, consulta [Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit](#).

Passaggio 4: Connect alla CodeCommit console e clona il repository

Se un amministratore ha già inviato il nome e i dettagli di connessione del repository CodeCommit, puoi ignorare questa fase e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.
 - Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
 - Copia l'URL HTTPS (GRC) se usi il comando git-remote-codecommit sul computer locale.
 - Copia l'URL SSH se utilizzi una coppia di public/private chiavi SSH con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in Regione AWS cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Apri un terminale. Dalla directory /tmp, esegui il comando git clone con l'URL SSH copiato per clonare il repository. Ad esempio, per clonare un repository denominato su un repository locale denominato *MyDemoRepo my-demo-repo* nella regione Stati Uniti orientali (Ohio):

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

Note

Se la verifica della connessione ha avuto esito positivo, a differenza del comando di clonazione, potresti non disporre dell'accesso richiesto al file config oppure un'altra impostazione potrebbe essere in conflitto con il file config. Prova a effettuare nuovamente la connessione, questa volta includendo nel comando l'ID della chiave SSH. Esempio:

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

Per ulteriori informazioni, consulta [Errore di accesso: la chiave pubblica viene caricata correttamente su IAM ma la connessione non riesce su sistemi Linux, macOS o Unix](#).

Per ulteriori informazioni su come connettersi ai repository, consulta [Connect al CodeCommit repository clonando il repository](#).

Fasi successive

Hai completato i prerequisiti. Segui i passaggi indicati per iniziare [Iniziare con CodeCommit](#) a utilizzare. CodeCommit

Passaggi di configurazione per le connessioni SSH ai AWS CodeCommit repository su Windows

Prima di poterti connettere AWS CodeCommit per la prima volta, devi completare alcuni passaggi iniziali di configurazione. Dopo aver configurato il computer e il AWS profilo, è possibile connettersi a un CodeCommit repository e clonarlo sul computer (operazione nota anche come creazione di un repository locale). Se non hai mai utilizzato Git, ti consigliamo di consultare le informazioni in [Dove posso saperne di più su Git?](#).

Argomenti

- [Fase 1: Configurazione iniziale per CodeCommit](#)
- [Fase 2: installazione di Git](#)
- [Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit](#)
- [Passaggio 4: Connect alla CodeCommit console e clona il repository](#)
- [Fasi successive](#)

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per configurare un account Amazon Web Services, creare un utente IAM e configurare l'accesso a CodeCommit.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la policy da allegare, scegli Avanti: Revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Note

Se desideri utilizzare i AWS CLI comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Fase 2: installazione di Git

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Se la versione di Git installata non include un emulatore Bash, ad esempio Git Bash, dovrai installarne uno. Per la configurazione delle connessioni SSH, utilizzerai questo emulatore anziché la riga di comando di Windows.

Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit

Per configurare le chiavi pubbliche e private per Git e CodeCommit su Windows

1. Apri l'emulatore Bash.

Note

Potrebbe essere necessario eseguire l'emulatore con autorizzazioni amministrative.

Dall'emulatore, esegui il comando `ssh-keygen` e segui le indicazioni per salvare il file nella directory `.ssh` del tuo profilo.

Esempio:

```
$ 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 . . . |
|           .       |
|                   |
|                   |
|                   |
+-----+

```

Questo genera:

- Il `codecommit_rsa` file, che è il file con chiave privata.

- Il `codecommit_rsa` file.pub, che è il file con la chiave pubblica.

Tip

Per impostazione predefinita, ssh-keygen genera una chiave a 2048 bit. È possibile utilizzare i parametri `-t` e `-b` per specificare il tipo e la lunghezza della chiave. Se desideri una chiave a 4096 bit nel formato rsa, devi specificarla eseguendo il comando con i seguenti parametri:

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

[Per ulteriori informazioni sui formati e le lunghezze richieste per le chiavi SSH, consulta Using IAM with. CodeCommit](#)

2. Esegui i seguenti comandi per visualizzare il valore del file della chiave pubblica (`codecommit_rsa.pub`):

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

Copia il contenuto del file, quindi chiudi Blocco note senza salvare. L'aspetto del contenuto del file è simile al seguente:

```
ssh-rsa EXAMPLE-AfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGFTYXpvaW5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE user-name@computer-name
```

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

Note

Puoi visualizzare e gestire direttamente le tue CodeCommit credenziali in Le mie credenziali di sicurezza. Per ulteriori informazioni, consulta [Visualizza e gestisci le tue credenziali](#).

4. Nella console IAM, nel riquadro di navigazione, scegli Utenti e dall'elenco degli utenti scegli il tuo utente IAM.
5. Nella pagina dei dettagli dell'utente, seleziona la scheda Security Credentials (Credenziali di sicurezza), quindi Upload SSH public key (Carica chiave pubblica SSH).
6. Incolla il contenuto della chiave pubblica SSH nel campo, quindi scegli Upload SSH public key (Carica chiave pubblica SSH).
7. Copia o salva le informazioni in SSH Key ID (ad esempio, *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
APKAEIBAERJR2EXAMPLE	2015-07-21 16:32 PDT	Active	Make Inactive Show SSH Key Delete

Note

Se hai IDs caricato più di una chiave SSH, le chiavi sono elencate alfabeticamente per ID chiave, non per data di caricamento. Assicurati di aver copiato l'ID della chiave associato alla data di caricamento corretta.

8. Nell'emulatore Bash, esegui i seguenti comandi per creare un file config nella directory ~/.ssh o per modificarne uno già esistente:

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

9. Aggiungi le seguenti righe al file, dove il valore per *User* è l'ID della chiave SSH che hai copiato in precedenza e il valore per *IdentityFile* è il percorso e il nome del file della chiave privata:

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

Note

Se hai assegnato al file della chiave privata un nome diverso da *codecommit_rsa*, assicurati di utilizzarlo qui.

Puoi configurare l'accesso SSH ai repository in più account Amazon Web Services. Per ulteriori informazioni, consulta. [Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit](#)

Salva il file come config (non config.txt), quindi chiudi Blocco note

 Important

Il nome del file deve essere config senza estensione file. In caso contrario, le connessioni SSH non vanno a buon fine.

10. Esegui il seguente comando per verificare la configurazione SSH:

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

Ti viene chiesto di confermare la connessione perché non `git-codecommit.us-east-2.amazonaws.com` è ancora inclusa nel tuo file hosts noto. L'impronta digitale del CodeCommit server viene visualizzata come parte della verifica (a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e per MD5 o 31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UES56fG6ZIZQ per SHA256).

 Note

CodeCommit le impronte digitali del server sono uniche per tutti. Regione AWS Per visualizzare le impronte digitali del server per un Regione AWS, vedere. [Impronte digitali del server per CodeCommit](#)

Dopo avere verificato la connessione, dovrebbe venire visualizzato un messaggio di conferma dell'avvenuta aggiunta del server ai file hosts noti e un messaggio dell'avvenuta connessione. Se non vedi un messaggio di successo, ricontrolla di aver salvato il config file nella directory `~/.ssh` dell'utente IAM a cui hai configurato l'accesso CodeCommit, che il config file non abbia l'estensione di file (ad esempio, non deve essere chiamato config.txt) e di aver specificato il file di chiave privata corretto (non .pub). `codecommit_rsa codecommit_rsa`

Per risolvere i problemi, esegui il comando con il parametro. `ssh -v` Esempio:

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

Per informazioni sulla risoluzione dei problemi di connessione, consulta [Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit](#).

Passaggio 4: Connect alla CodeCommit console e clona il repository

Se un amministratore ha già inviato il nome e i dettagli di connessione del repository CodeCommit, puoi ignorare questa fase e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.
 - Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
 - Copia l'URL HTTPS (GRC) se usi il comando git-remote-codecommit sul computer locale.
 - Copia l'URL SSH se utilizzi una coppia di public/private chiavi SSH con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in Regione AWS cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Nell'emulatore Bash, eseguire il comando git clone con l'URL SSH copiato per clonare il repository. Questo comando crea il repository locale in una sottodirectory della directory in cui

viene eseguito il comando. Ad esempio, per clonare un repository denominato su un repository locale denominato *MyDemoRepo my-demo-repo* nella regione Stati Uniti orientali (Ohio):

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

In alternativa, apri un prompt dei comandi ed esegui il comando utilizzando l'URL e l'ID della chiave SSH per la chiave pubblica che hai caricato su IAM. git clone Il repository locale viene creato in una sottodirectory della directory in cui viene eseguito il comando. Ad esempio, per clonare il repository *MyDemoRepo* nel repository locale *my-demo-repo*:

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

Per ulteriori informazioni, consultare [Connect al CodeCommit repository clonando il repository](#) e [Crea un commit](#).

Fasi successive

Hai completato i prerequisiti. Segui i passaggi indicati [Iniziare con CodeCommit](#) per iniziare a utilizzare. CodeCommit

Passaggi di configurazione per le connessioni HTTPS ai AWS CodeCommit repository su Linux, macOS o Unix con l'helper delle credenziali AWS CLI

Prima di poterti connettere AWS CodeCommit per la prima volta, devi completare i passaggi di configurazione iniziali. Per la maggior parte degli utenti, ciò può essere eseguito più facilmente attenendosi alle fasi descritte in [Per utenti HTTPS che utilizzano credenziali Git](#). Tuttavia, se desideri connetterti CodeCommit utilizzando un account root, un accesso federato o credenziali temporanee, puoi utilizzare l'helper per le credenziali incluso in. AWS CLI

Note

Sebbene l'helper per le credenziali sia un metodo supportato per la connessione CodeCommit tramite accesso federato, un provider di identità o credenziali temporanee, il

metodo consigliato consiste nell'installare e utilizzare l'utilità. `git-remote-codecommit` Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Argomenti

- [Fase 1: Configurazione iniziale per CodeCommit](#)
- [Fase 2: installazione di Git](#)
- [Passaggio 3: configura l'helper delle credenziali](#)
- [Passaggio 4: Connect alla CodeCommit console e clona il repository](#)
- [Passaggi successivi](#)

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per configurare un account Amazon Web Services, creare e configurare un utente IAM e installare il AWS CLI.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da. CodeCommit Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>

4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la politica da allegare, scegli Avanti: revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Per installare e configurare AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con `git-remote-codecommit`

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o esegui l'aggiornamento AWS CLI all'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando.

Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il configure comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS dove si trova il repository, ad esempio. us-east-2 Quando viene richiesto il formato di output predefinito, specificare json. Ad esempio, se stai configurando un profilo per un utente 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
```

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Fase 2: installazione di Git

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Passaggio 3: configura l'helper delle credenziali

1. Dal terminale, usa Git per eseguire `git config`, specificando l'uso dell'helper per le credenziali Git con il profilo delle AWS credenziali e abilitando l'helper per le credenziali Git per inviare il percorso ai repository:

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

Tip

L'helper delle credenziali utilizza il profilo di AWS credenziali predefinito o il ruolo dell'istanza Amazon EC2. È possibile specificare un profilo da utilizzare, ad esempio `CodeCommitProfile`, se è stato creato un profilo di credenziali AWS per l'uso con CodeCommit:

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

Se il nome del profilo contiene spazi, assicurati di inserire il nome tra virgolette (").

Puoi configurare i profili per repository specifici utilizzando `--local` anziché `--global`.

L'assistente credenziali Git scrive il seguente valore in `~/.gitconfig`:

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

 Important

Se desideri utilizzare un utente IAM diverso sullo stesso computer locale per CodeCommit, devi eseguire nuovamente il git config comando e specificare un profilo di credenziali diverso AWS .

2. Eseguire `git config --global --edit` per verificare che il valore precedente sia stato scritto in `~/.gitconfig`. In caso di esito positivo, viene visualizzato il valore precedente (in aggiunta ai valori che potrebbero essere già presenti nel file di configurazione globale Git). Per uscire, in genere è sufficiente digitare `:q` e premere Invio.

In caso di problemi successivi alla configurazione dell'assistente credenziali, consulta [Risoluzione dei problemi](#).

 Important

Se utilizzi macOS, segui i passaggi seguenti per assicurarti che l'helper delle credenziali sia configurato correttamente.

3. Se usi macOS, usa HTTPS per [connetterti a un CodeCommit](#) repository. Dopo la prima connessione a un CodeCommit repository con HTTPS, l'accesso successivo fallisce dopo circa 15 minuti. La versione Git predefinita su macOS utilizza l'utilità Keychain Access per archiviare le credenziali. Per quanto riguarda le misure di sicurezza, la password generata per l'accesso al CodeCommit repository è temporanea, quindi le credenziali memorizzate nel portachiavi smettono di funzionare dopo circa 15 minuti. Per impedire l'utilizzo delle credenziali scadute, è necessario:
 - Installare una versione di Git che non utilizza il portachiavi per impostazione predefinita.
 - Configurare l'utilità Keychain Access per non fornire credenziali per gli archivi. CodeCommit

1. Aprire l'utilità Accesso Portachiavi. È possibile utilizzare Finder per individuarla.
2. Cercare `git-codecommit.us-east-2.amazonaws.com`. Evidenziare la riga, aprire il menu contestuale o fare clic con il pulsante destro del mouse su di esso e scegliere Ottieni informazioni.
3. Selezionare la scheda Controllo accesso.
4. In Conferma prima di consentire l'accesso scegliere `git-credential-osxkeychain` e quindi scegliere il segno meno per rimuovere la voce dall'elenco.

Note

Dopo avere rimosso `git-credential-osxkeychain` dall'elenco, ogni volta che si esegue un comando Git viene visualizzato un messaggio a comparsa. Scegliere Nega per continuare. Se la finestra di dialogo popup provoca problemi, ecco alcune altre soluzioni:

- Connect CodeCommit tramite SSH anziché HTTPS. Per ulteriori informazioni, consulta [Per connessioni SSH su Linux, macOS o Unix](#).
- Nell'utilità Accesso Portachiavi, nella scheda Controllo Accesso per `git-codecommit.us-east-2.amazonaws.com`, scegliere l'opzione Consenti a tutte le applicazioni di accedere a questo elemento (l'accesso a questo elemento non è limitato). Le finestre di dialogo popup non vengono visualizzate, ma le credenziali scadranno, in media, ogni 15 minuti circa e viene visualizzato un messaggio di errore 403. A quel punto, è necessario eliminare l'elemento portachiavi per ripristinare la funzionalità.
- Per ulteriori informazioni, consulta [Git per macOS: ho configurato correttamente l'helper delle credenziali, ma ora mi viene negato l'accesso al mio repository \(403\)](#).

Passaggio 4: Connect alla CodeCommit console e clona il repository

Se un amministratore ti ha già inviato il nome e i dettagli di connessione per il CodeCommit repository, puoi saltare questo passaggio e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.
 - Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
 - Copia l'URL HTTPS (GRC) se usi il comando git-remote-codecommit sul computer locale.
 - Copia l'URL SSH se utilizzi una coppia di chiavi SSH pubblica/privata con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Aprire un terminale ed eseguire il comando git clone con l'URL HTTPS copiato. Ad esempio, per clonare un repository denominato su un repository locale denominato *MyDemoRepo my-demo-repo* nella regione Stati Uniti orientali (Ohio):

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

Passaggi successivi

Hai completato i prerequisiti. Segui i passaggi indicati per iniziare [Iniziare con CodeCommit](#) a utilizzare. CodeCommit

Passaggi di configurazione per le connessioni HTTPS ai AWS CodeCommit repository su Windows con l'helper delle AWS CLI credenziali

Prima di poterti connettere AWS CodeCommit per la prima volta, devi completare i passaggi di configurazione iniziali. Per la maggior parte degli utenti, ciò può essere eseguito più facilmente attenendosi alle fasi descritte in [Per utenti HTTPS che utilizzano credenziali Git](#). Tuttavia, se desideri connetterti CodeCommit utilizzando un account root, un accesso federato o credenziali temporanee, puoi utilizzare l'helper per le credenziali incluso in AWS CLI

Note

Sebbene l'helper per le credenziali sia un metodo supportato per la connessione CodeCommit tramite accesso federato, un provider di identità o credenziali temporanee, il metodo consigliato consiste nell'installare e utilizzare l'utilità. `git-remote-codecommit` Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Questo argomento illustra i passaggi per installare AWS CLI, configurare il computer e il AWS profilo, connettersi a un CodeCommit repository e clonare tale repository sul computer, operazione nota anche come creazione di un repository locale. Se non hai mai utilizzato Git, ti consigliamo di consultare le informazioni in [Dove posso saperne di più su Git?](#).

Argomenti

- [Fase 1: Configurazione iniziale per CodeCommit](#)
- [Fase 2: installazione di Git](#)
- [Passaggio 3: configura l'helper delle credenziali](#)
- [Passaggio 4: Connect alla CodeCommit console e clona il repository](#)
- [Passaggi successivi](#)

Fase 1: Configurazione iniziale per CodeCommit

Segui questi passaggi per configurare un account Amazon Web Services, creare e configurare un utente IAM e installare il AWS CLI. AWS CLI Include un supporto per le credenziali che puoi configurare per le connessioni HTTPS ai tuoi CodeCommit repository.

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che all'utente non siano associate policy che neghino espressamente le AWS KMS azioni richieste da. CodeCommit Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUsero un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la politica da allegare, scegli Avanti: revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Per installare e configurare AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con. git-remote-codecommit

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

 Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o esegui l'aggiornamento AWS CLI all'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando.

Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il `configure` comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS

dove si trova il repository, ad esempio. us-east-2 Quando viene richiesto il formato di output predefinito, specificare json. Ad esempio, se stai configurando un profilo per un utente 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
```

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Fase 2: installazione di Git

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti Web come [Git per Windows](#). Se usi questo link per installare Git, puoi accettare tutte le impostazioni predefinite di installazione tranne le seguenti:

- Quando richiesto durante il passaggio Adjusting your PATH environment, scegli l'opzione per usare Git dalla riga di comando.

- (Facoltativo) Se intendi utilizzare HTTPS con l'helper per le credenziali incluso in AWS CLI invece di configurare le credenziali Git CodeCommit per, nella pagina Configurazione delle opzioni extra, assicurati che l'opzione Enable Git Credential Manager sia deselezionata. Git Credential Manager è compatibile solo CodeCommit se gli utenti IAM configurano le credenziali Git. Per ulteriori informazioni, consulta [Per utenti HTTPS che utilizzano credenziali Git](#) e [Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository \(403\)](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

Passaggio 3: configura l'helper delle credenziali

AWS CLI Include un aiutante per le credenziali Git con cui puoi usare. CodeCommit L'helper per le credenziali Git richiede un profilo di AWS credenziali, che memorizza una copia dell'ID della chiave di accesso e della chiave di AWS accesso AWS segreta di un utente IAM (insieme a un Regione AWS nome e un formato di output predefiniti). L'helper per le credenziali Git utilizza queste informazioni per l'autenticazione automatica, CodeCommit quindi non è necessario inserire queste informazioni ogni volta che si utilizza Git per interagire con. CodeCommit

1. Apri un prompt dei comandi e usa Git per eseguire `git config`, specificando l'uso dell'helper di credenziali Git con il profilo delle AWS credenziali, che consente all'helper di credenziali Git di inviare il percorso ai repository:

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

L'assistente credenziali Git scrive il seguente valore nel file `.gitconfig`.

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

⚠ Important

- Se stai utilizzando un emulatore Bash anziché la riga di comando di Windows, dovrai utilizzare le virgolette singole anziché quelle doppie.
- L'helper delle credenziali utilizza il AWS profilo predefinito o il ruolo dell' EC2 istanza Amazon. Se hai creato un profilo di AWS credenziali da utilizzare, ad esempio *CodeCommitProfile*, puoi modificare il comando come segue per utilizzarlo invece:

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

Nel file .gitconfig viene scritto quanto segue:

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

- Se il nome del profilo include spazi, dovrai modificare il file .gitconfig dopo avere eseguito il comando per includerlo tra virgolette singole ('). In caso contrario, l'assistente credenziali non funzionerà.
- Se l'installazione di Git per Windows includeva l'utilità Git Credential Manager, vengono visualizzati errori 403 o richieste di specifica delle credenziali nell'utilità Credential Manager dopo i primi tentativi di connessione. Il modo più sicuro per risolvere questo problema è disinstallare e reinstallare Git per Windows senza l'opzione per l'utilità Git Credential Manager, poiché non è compatibile con CodeCommit. Se si desidera mantenere l'utilità Git Credential Manager, è necessario eseguire ulteriori passaggi di configurazione da utilizzare CodeCommit, inclusa la modifica manuale del file.gitconfig per specificare l'uso dell'helper delle credenziali per la connessione a. AWS CodeCommit CodeCommit Rimuovi eventuali credenziali archiviate dall'utilità Credential Manager (disponibile nel Pannello di controllo). Dopo avere rimosso le credenziali archiviate, aggiungi quanto segue al file .gitconfig, salva il file ed effettua una nuova connessione da una nuova finestra del prompt dei comandi:

```
[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
```

Potrebbe anche essere necessario riconfigurare le impostazioni git config specificando `--system` anziché `--global` o `--local` perché tutte le connessioni funzionino come previsto.

- Se desideri utilizzare utenti IAM diversi sullo stesso computer locale per CodeCommit, devi specificare `git config --local` invece di `ed` eseguire la configurazione per ogni profilo di `git config --global` credenziali. AWS

2. Esegui `git config --global --edit` per verificare che i valori precedenti siano stati scritti nel file `.gitconfig` per il profilo utente (per impostazione predefinita, `%HOME%\.gitconfig` oppure `drive:\Users\UserName\.gitconfig`). In caso di esito positivo, vengono visualizzati i valori precedenti (in aggiunta ai valori che potrebbero essere già presenti nel file di configurazione globale Git). Per uscire, in genere è sufficiente digitare `:q` e premere Invio.

Passaggio 4: Connect alla CodeCommit console e clona il repository

Se un amministratore ti ha già inviato il nome e i dettagli di connessione per il CodeCommit repository, puoi saltare questo passaggio e clonare direttamente il repository.

Per connettersi a un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.

- Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
- Copia l'URL HTTPS (GRC) se usi il comando `git-remote-codecommit` sul computer locale.
- Copia l'URL SSH se utilizzi una coppia di chiavi pubblica/privata SSH con il tuo utente IAM.

Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. Apri un prompt dei comandi ed esegui il `git clone` comando con l'URL HTTPS che hai copiato. Il repository locale viene creato in una sottodirectory della directory in cui viene eseguito il comando. Ad esempio, per clonare un repository denominato in un repository locale denominato *MyDemoRepo my-demo-repo* nella regione Stati Uniti orientali (Ohio):

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

In alcune versioni di Windows, potrebbe venire visualizzato un messaggio popup con la richiesta di nome utente e password. Questo è il sistema di gestione delle credenziali integrato per Windows, che tuttavia non è compatibile con l'assistente credenziali per AWS CodeCommit. Seleziona Annulla.

Passaggi successivi

Hai completato i prerequisiti. Segui i passaggi indicati per iniziare [Iniziare con CodeCommit](#) a utilizzare. CodeCommit

Nozioni di base

Il modo più semplice per iniziare CodeCommit è seguire i passaggi riportati di seguito. [Iniziare con CodeCommit](#) Se non conosci Git e CodeCommit dovresti anche prendere in considerazione la possibilità di seguire i passaggi indicati [Guida introduttiva a Git e CodeCommit](#). Questo ti aiuta a familiarizzare con Git CodeCommit e ad acquisire le nozioni di base sull'uso di Git quando interagisci con i tuoi repository. CodeCommit

Puoi anche seguire il tutorial in [Simple Pipeline Walkthrough CodePipeline](#) e imparare CodeCommit a usare il tuo CodeCommit repository come parte di una pipeline di distribuzione continua.

I tutorial in questa sezione presuppongono che siano stati completati i [prerequisiti e la configurazione](#), tra cui:

- Assegnazione delle autorizzazioni all'utente IAM.
- Configurazione della gestione delle credenziali per le connessioni HTTPS o SSH sul computer locale in uso per questo tutorial.
- Configurazione di AWS CLI se si desidera utilizzare la riga di comando o il terminale per tutte le operazioni, inclusa la creazione del repository.

Argomenti

- [Iniziare con AWS CodeCommit](#)
- [Guida introduttiva a Git e AWS CodeCommit](#)

Iniziare con AWS CodeCommit

Questo tutorial mostra come utilizzare alcune CodeCommit funzionalità chiave. Innanzitutto, crea un repository e apporta alcune modifiche. Quindi, puoi esplorare i file e visualizzare le modifiche. Potrai anche creare una richiesta di pull per consentire ad altri utenti di esaminare e commentare le modifiche al codice.

Se desideri migrare il codice esistente in CodeCommit, consulta [Migrare a AWS CodeCommit](#).

Se non hai familiarità con Git, considera di completare anche [Guida introduttiva a Git e CodeCommit](#). Dopo aver completato questi tutorial, dovresti avere abbastanza pratica per iniziare a utilizzarli CodeCommit per i tuoi progetti e in ambienti di team.

La CodeCommit console include informazioni utili in un pannello pieghevole che puoi aprire dall'icona delle informazioni



o da qualsiasi link Info sulla pagina. Puoi chiudere il pannello in qualsiasi momento.

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
7     references.ref;});
8     console.log('References:', references);
9
10    //Get the repository from the event and show its git clone URL
11    var repository = event.Records[0].eventSourceARN.split(":")[5];
12    var params = {
13        repositoryName: repository
14    };
15    codecommit.getRepository(params, function(err, data) {
16        if (err) {
17            console.log(err);
18            var message = "Error getting repository metadata for repository " + repository;
19            console.log(message);
20            context.fail(message);
21        } else {
22            console.log('Clone URL', data.repositoryMetadata.cloneUrlHttp);
23            context.succeed(data.repositoryMetadata.cloneUrlHttp);
24        }
25    });
26 }

```

View a file in a repository

You can view the contents of most file types in the console. For Markdown files, you can switch between a rendered and source code view. For binary files, you must confirm before the file content is displayed.

If you want to quickly make a change to the file, you can edit it in the console.

Not all files can be displayed in the console. For example, if the file size exceeds the commit object limit, you cannot view it in the console. You must view it in a local repo.

Learn more

- Browse files
- Edit a file
- Add a file

La CodeCommit console offre anche un modo per cercare rapidamente le risorse, come repository, progetti di compilazione, applicazioni di distribuzione e pipeline. Scegli Go to resource (Vai alla risorsa) o premi il tasto / e digita il nome della risorsa. Qualsiasi corrispondenza verrà visualizzata nell'elenco. Le ricerche rispettano la distinzione tra maiuscole e minuscole. Puoi visualizzare solo le risorse per le quali disponi dell'autorizzazione di visualizzazione. Per ulteriori informazioni, consulta [Visualizzazione di risorse nella console](#).

Prerequisiti

Prima di iniziare, devi completare i [prerequisiti e la configurazione](#), che includono:

- Assegnazione delle autorizzazioni all'utente IAM.

- Impostazione della gestione delle credenziali per le connessioni HTTPS o SSH sul computer locale utilizzato per questo tutorial.
- Configurazione di AWS CLI se si desidera utilizzare la riga di comando o il terminale per tutte le operazioni, inclusa la creazione del repository.

Argomenti

- [Fase 1: creazione di un repository CodeCommit](#)
- [Passaggio 2: aggiungi file al tuo repository](#)
- [Passaggio 3: sfoglia il contenuto del tuo repository](#)
- [Passaggio 4: Crea e collabora su una pull request](#)
- [Fase 5: rimozione](#)
- [Fase 6: fasi successive](#)

Fase 1: creazione di un repository CodeCommit

È possibile utilizzare la CodeCommit console per creare un CodeCommit repository. Se disponi già di un repository che intendi utilizzare per questo tutorial, puoi ignorare questa fase.

Note

A seconda dell'utilizzo, potrebbero esserti addebitati costi per la creazione o l'accesso a un repository. Per ulteriori informazioni, consulta la sezione [Prezzi](#) nella pagina di informazioni sul CodeCommit prodotto.

Per creare il CodeCommit repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Usa il selettore della regione per scegliere Regione AWS dove vuoi creare il repository. Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository), scegli Create repository (Crea repository).
4. Nella pagina Create repository (Crea repository), in Repository name (Nome repository), immetti un nome per il repository (ad esempio **MyDemoRepo**).

 Note

I nomi dei repository distinguono tra lettere maiuscole e minuscole e non possono superare i 100 caratteri. Per ulteriori informazioni, consulta [Limiti](#).

5. (Facoltativo) In Description (Descrizione), immettere una descrizione, ad esempio **My demonstration repository**. Ciò può consentire agli utenti di identificare lo scopo del repository.
6. (Facoltativo) Scegliete Aggiungi tag per aggiungere uno o più tag del repository (un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse) al repository. Per ulteriori informazioni, consulta [Taggare i repository in AWS CodeCommit](#).
7. (Facoltativo) Espandi la configurazione aggiuntiva per specificare se utilizzare la chiave predefinita Chiave gestita da AWS o la chiave gestita dal cliente per crittografare e decrittografare i dati in questo repository. Se scegli di utilizzare la tua chiave gestita dal cliente, devi assicurarti che sia disponibile nel luogo in Regione AWS cui stai creando il repository e che la chiave sia attiva. Per ulteriori informazioni, consulta [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#).
8. (Facoltativo) Seleziona Abilita Amazon CodeGuru Reviewer per Java e Python se questo repository conterrà codice Java o Python e desideri che Reviewer analizzi CodeGuru quel codice. CodeGuru Reviewer utilizza diversi modelli di machine learning per trovare i difetti del codice e suggerire automaticamente miglioramenti e correzioni nelle pull request. Per ulteriori informazioni, consulta la Amazon CodeGuru Reviewer User Guide.
9. Scegli Create (Crea) .

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

Se utilizzi un nome diverso da MyDemoRepo per il repository, accertati di utilizzarlo nelle fasi rimanenti di questo tutorial.

All'apertura del repository, vengono visualizzate informazioni su come aggiungere file direttamente dalla CodeCommit console.

Passaggio 2: aggiungi file al tuo repository

Puoi aggiungere file al repository:

- Creazione di un file nella CodeCommit console. Se crei il primo file per un repository nella console, viene creato per te un ramo chiamato main. Questo ramo è il ramo predefinito per il tuo repository.
- Caricamento di un file dal computer locale tramite la CodeCommit console. Se carichi il primo file per un repository dalla console, viene creato un ramo chiamato main. Questo ramo è il ramo predefinito per il tuo repository.
- Usare un client Git per clonare il repository sul computer locale e quindi aggiungere, eseguire il commit e inviare file al repository. CodeCommit Un ramo viene creato per te come parte del primo commit da Git ed è impostato come ramo predefinito per il tuo repository. Il nome del ramo è la scelta predefinita del tuo client Git. Prendi in considerazione la possibilità di configurare il tuo client Git per utilizzare main come nome per il ramo iniziale.

Note

Puoi creare rami e modificare il ramo predefinito per un repository in qualsiasi momento. Per ulteriori informazioni, consulta [Lavorare con le filiali nei repository AWS CodeCommit](#).

Il modo più semplice per iniziare è aprire la CodeCommit console e aggiungere un file. In questo modo, crei anche un ramo predefinito per il tuo repository chiamato main. Per istruzioni su come aggiungere un file e creare un primo commit in un repository utilizzando il AWS CLI, consulta [Creare il primo commit per un repository](#) utilizzando il. AWS CLI

Per aggiungere un file al repository

1. Nella barra di navigazione per il repository, scegliere Code (Codice).
2. Scegliere Add file (Aggiungi file) e quindi scegliere di creare un file o di caricarne uno dal computer. Questo tutorial mostra come fare entrambe le cose.
3. Per aggiungere un file, eseguire le operazioni seguenti:
 - a. Nell'elenco a discesa dei rami, scegliere il ramo in cui si desidera aggiungere il file. Il ramo predefinito viene selezionato automaticamente. Nell'esempio mostrato qui, il ramo predefinito è denominato *main*. Se si desidera aggiungere il file a un ramo diverso, scegliere un ramo diverso.

- b. In File name (Nome file), immettere un nome per il file. Nell'editor di codice, immettere codice per il file.
- c. In Author name (Nome autore), immettere il nome che si desidera mostrare agli altri utenti del repository.
- d. In Email address (Indirizzo e-mail), immettere l'indirizzo e-mail.
- e. (Facoltativo) In Commit message (Messaggio di commit), immettere un breve messaggio. Sebbene questa operazione sia facoltativa, consigliamo vivamente di aggiungere un messaggio di commit per aiutare i membri del team a comprendere il motivo dell'aggiunta del file. Se non si immette un messaggio di commit, viene utilizzato un messaggio predefinito.
- f. Scegliere Commit changes (Applica modifiche).

Per caricare un file, procedere come segue:

- Se si carica un file, scegliere il file da caricare.

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](#)

- In Author name (Nome autore), immettere il nome che si desidera mostrare agli altri utenti del repository.
- In Email address (Indirizzo e-mail), immettere l'indirizzo e-mail.
- (Facoltativo) In Commit message (Messaggio di commit), immettere un breve messaggio. Sebbene questa operazione sia facoltativa, consigliamo vivamente di aggiungere un

messaggio di commit per aiutare i membri del team a comprendere il motivo dell'aggiunta del file. Se non si immette un messaggio di commit, viene utilizzato un messaggio predefinito.

- Scegliere Commit changes (Applica modifiche).

Per ulteriori informazioni, consulta [Lavorare con i file nei AWS CodeCommit repository](#).

Per usare un client Git per clonare il repository, installa Git sul tuo computer locale, quindi clona il repository. CodeCommit Aggiungi alcuni file al repository locale ed esegui il push al repository CodeCommit . Per un'introduzione approfondita, segui il [Guida introduttiva a Git e CodeCommit](#). Se hai familiarità con Git, ma non sei sicuro di come farlo con un CodeCommit repository, puoi visualizzare esempi e istruzioni in [Crea un commitFase 2: Creare un repository locale](#), o [Connessione a un repository](#).

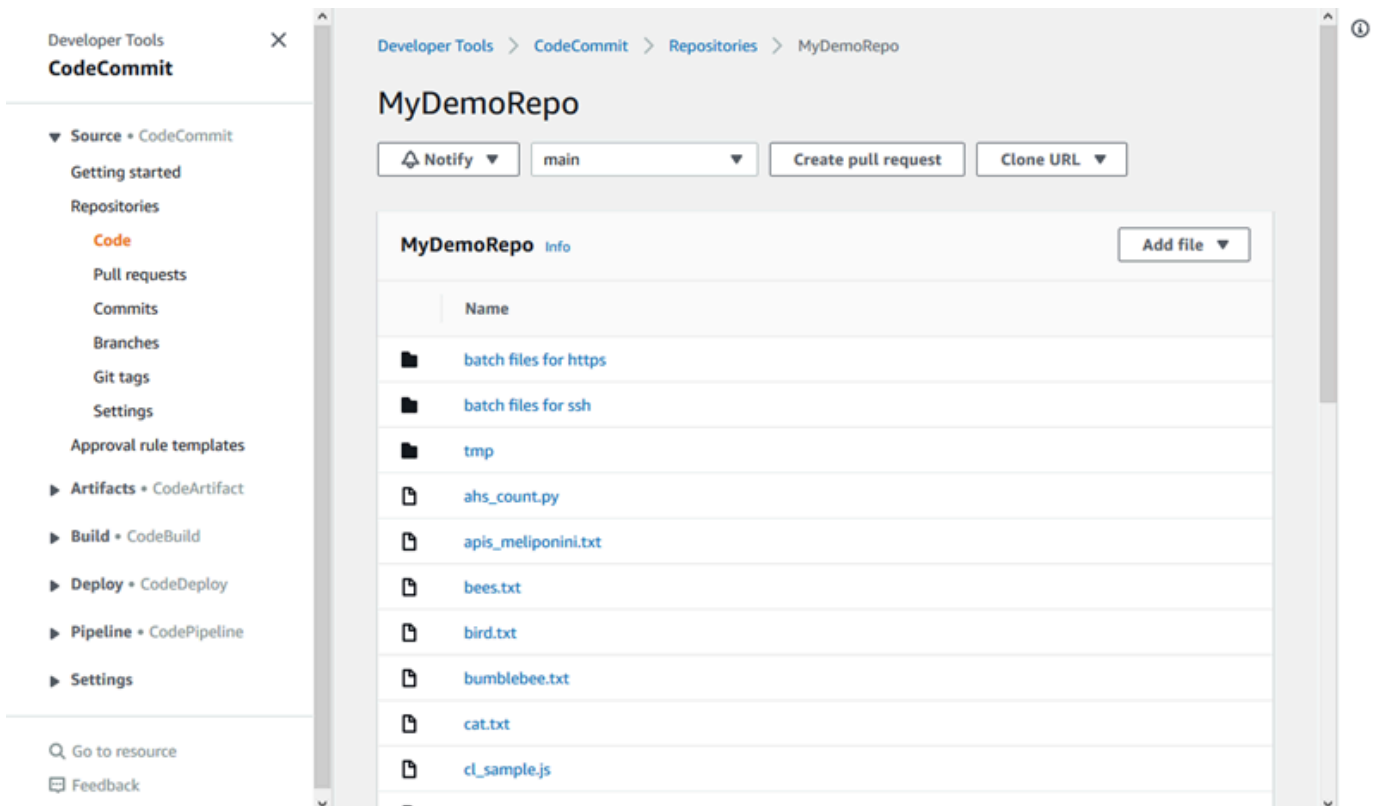
Dopo aver aggiunto alcuni file al CodeCommit repository, puoi visualizzarli nella console.

Passaggio 3: sfoglia il contenuto del tuo repository

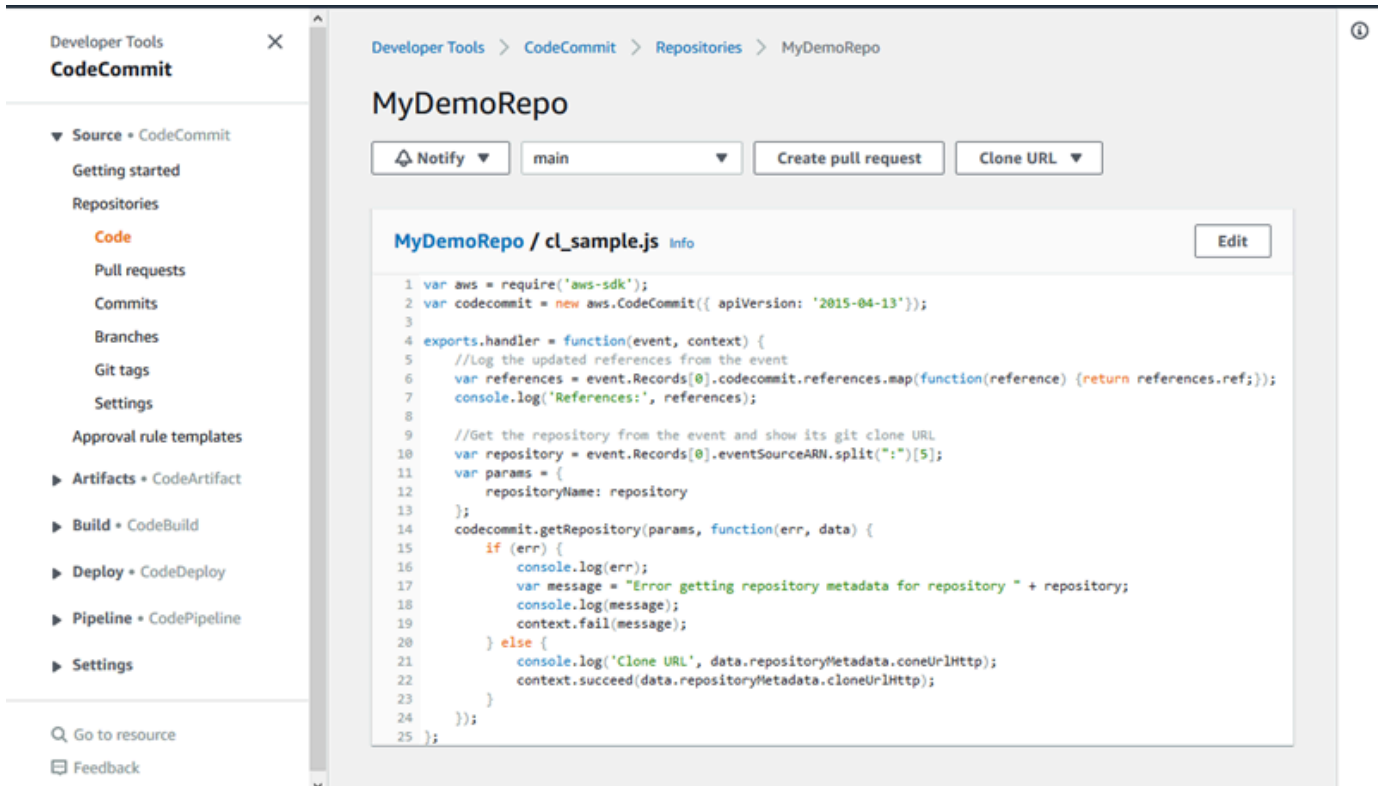
Puoi usare la CodeCommit console per esaminare i file in un repository o leggere rapidamente il contenuto di un file. È utile per determinare il ramo da controllare o se creare una copia locale di un repository.

Per sfogliare il repository

1. Da Repositories, scegli. MyDemoRepo
2. La pagina visualizza il contenuto nel ramo predefinito del repository. Per visualizzare un altro ramo o in grado di mostrare il codice su un tag specifico, scegliere il ramo o il tag che si desidera visualizzare dall'elenco. Nella schermata seguente, la vista è impostata sul ramo principale.

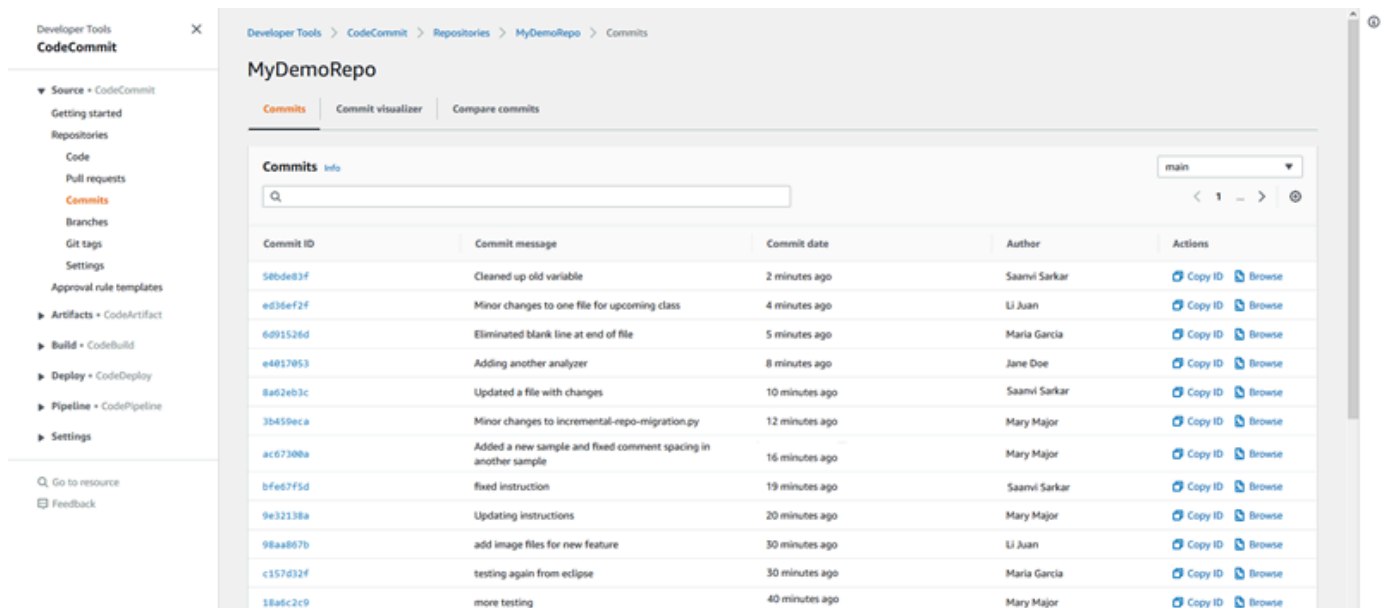


3. Per visualizzare il contenuto di un file nel repository, scegliere il file dall'elenco. Per cambiare il colore del codice visualizzato, scegliere l'icona delle impostazioni.



Per ulteriori informazioni, consulta [Sfogliare i file in un repository](#).

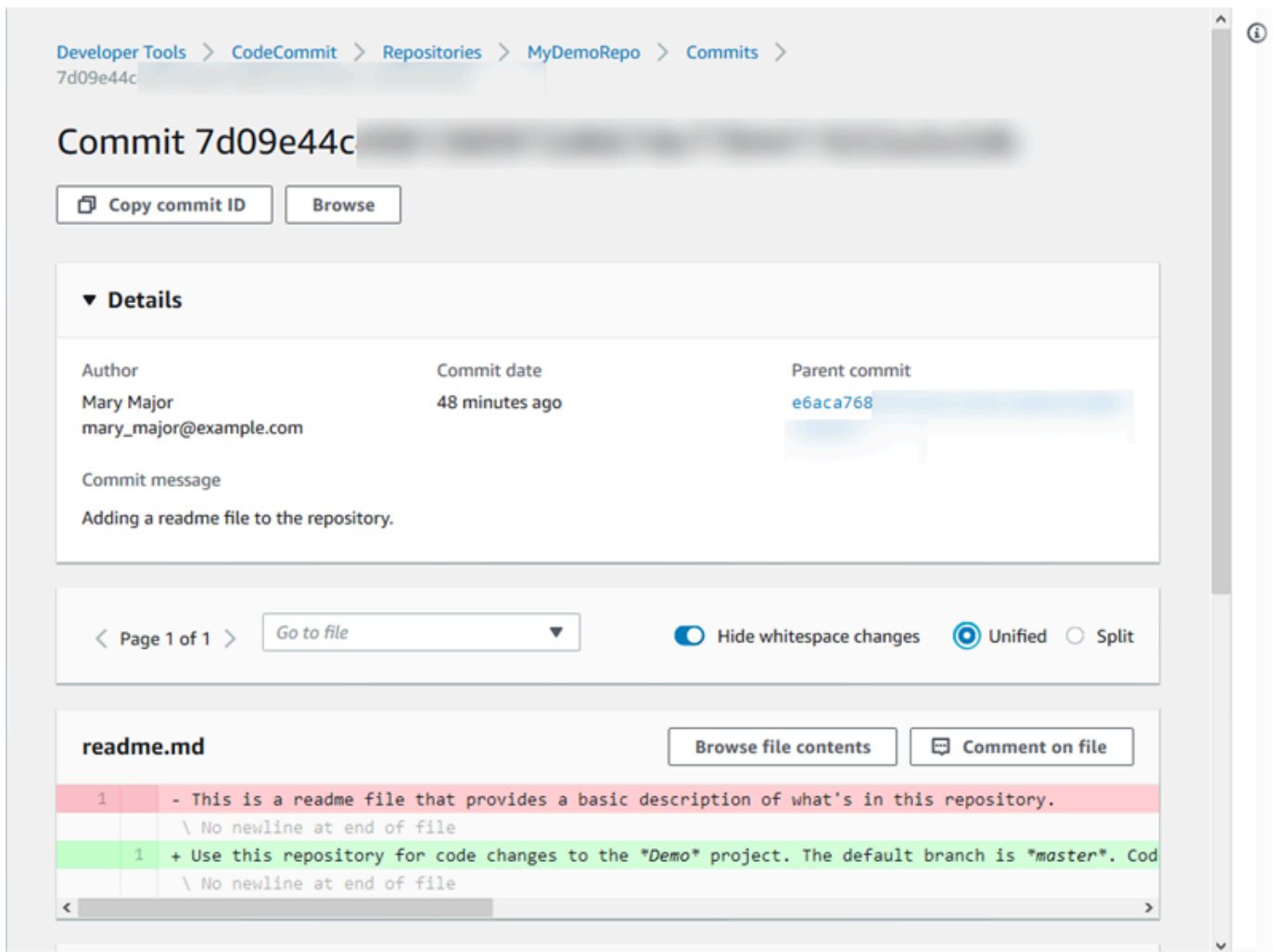
4. Per sfogliare la cronologia dei commit del repository, scegliere Commit. La console visualizza la cronologia dei commit per il ramo predefinito, in ordine cronologico inverso. Rivedere i dettagli del commit per autore, data e altro ancora.



5. Per visualizzare la cronologia dei commit per [ramo](#) o per [tag Git](#), scegliere il ramo o il tag che si desidera visualizzare dall'elenco.
6. Per vedere le differenze tra un commit e il relativo commit padre, scegliere l'ID commit abbreviato. È possibile scegliere in che modo visualizzare le modifiche, ad esempio se mostrare o nascondere le modifiche agli spazi vuoti e se visualizzare le modifiche inline (visualizzazione unificata) o affiancate (visualizzazione doppia).

Note

Le preferenze per la visualizzazione del codice e per altre impostazioni della console vengono salvate come cookie del browser a ogni modifica. Per ulteriori informazioni, consulta [Utilizzo delle preferenze dell'utente](#).



7. Per visualizzare tutti i commenti su un commit, scegliere il commit e quindi scorrere le modifiche per visualizzarle inline. È anche possibile aggiungere commenti propri e rispondere ai commenti altrui.

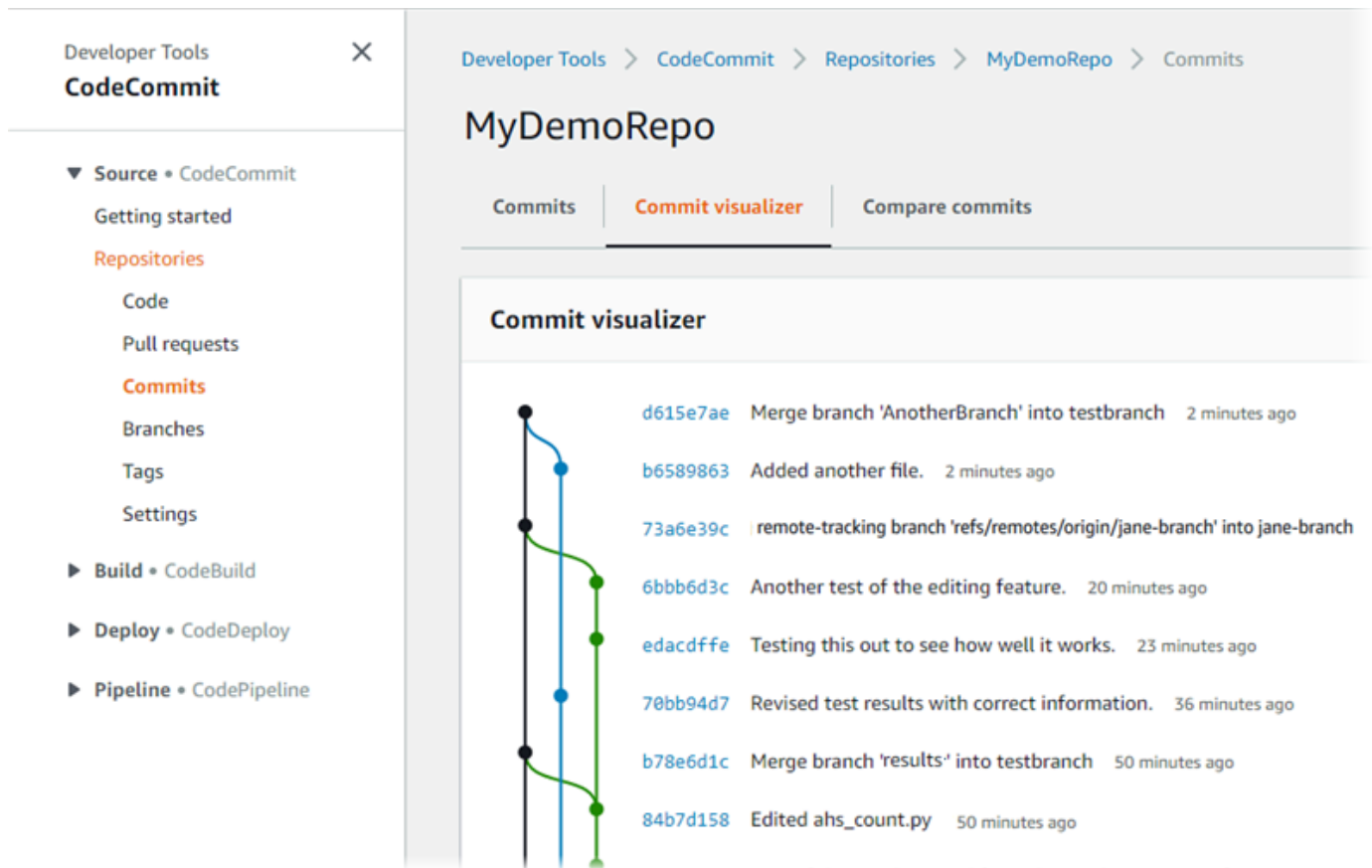
Per ulteriori informazioni, consulta [Commentare un commit](#).

8. Per visualizzare le differenze tra due specificatori di commit, inclusi tag, branch e commit, nel riquadro di navigazione IDs, scegli Commit, quindi scegli Confronta commit.

The screenshot shows the 'Compare' page in the AWS CodeCommit console for the repository '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', with the last one being the active tab. The 'Destination' dropdown is set to 'AnotherBranch' and the 'Source' dropdown is set to '6b65eb76'. There are 'Compare' and 'Cancel' buttons. Below this, there is a 'Page 1 of 1' indicator, a 'Go to file' dropdown, and three toggle options: '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. The diff highlights changes: a red line for a removed print statement and a green line for an added print statement. The code includes a comment about using specific texts like 'bumblebee.txt'. Below the diff, there is a section for 'anothernew/dir2/anothertest.txt' marked as 'Added', with 'Browse file contents' and 'Comment on file' buttons.

Per ulteriori informazioni, consulta [Sfogliare la cronologia dei commit di un repository](#) e [Confrontare i commit](#).

9. In Commits (Commit), scegliere la scheda Commit visualizer (Visualizzatore commit).



Il grafico del commit viene visualizzato con la riga dell'oggetto per ogni commit mostrata accanto al relativo punto sul grafico. La visualizzazione della riga dell'oggetto è limitata a 80 caratteri.

10. Per visualizzare ulteriori dettagli su un commit, scegliere il relativo ID abbreviato. Per eseguire il rendering del grafico da uno specifico commit, scegliere il relativo punto nel grafico. Per ulteriori informazioni, consulta [Visualizza un grafico della cronologia dei commit di un repository](#).

Passaggio 4: Crea e collabora su una pull request

Quando si lavora in un repository con altri utenti, è possibile collaborare al codice e rivedere le modifiche. È possibile creare una richiesta pull in modo che altri utenti possano rivedere e commentare le modifiche al codice in un ramo. È inoltre possibile creare una o più regole di approvazione per la richiesta pull. Ad esempio, è possibile creare una regola di approvazione che richiede che almeno altri due utenti approvino la richiesta pull prima che possa essere unita. Dopo l'approvazione della richiesta pull, è possibile unire tali modifiche nel relativo ramo di destinazione. Se si impostano le notifiche per il repository, i relativi utenti possono ricevere e-mail su eventi relativi al repository (ad esempio le richieste pull o quando qualcuno commenta il codice). Per

ulteriori informazioni, consulta [Configurazione delle notifiche per gli eventi in un repository AWS CodeCommit](#).

 Important

Prima di poter creare una richiesta di pull, è necessario creare un ramo che contenga le modifiche al codice che si desidera esaminare. Per ulteriori informazioni, consulta [Creare un ramo](#).

Per creare e collaborare su una richiesta di pull

1. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull).
2. In Pull request (richiesta di pull), scegliere Create pull request (Crea richiesta di pull).

 Tip

È anche possibile creare richieste pull da Branches (Rami) e Code (Codice).

In Create pull request (Crea richiesta di pull), in Source (Origine), scegliere il ramo che contiene le modifiche da far esaminare. In Destination (Destinazione), scegliere il ramo in cui si desidera unire il codice riesaminato quando la richiesta di pull viene chiusa. Scegliere Compare (Confronta).

3. Esaminare i dettagli dell'unione e le modifiche per confermare che la richiesta di pull contenga le modifiche e i commit desiderati. In caso affermativo, in Title (Titolo), immettere un titolo per questa revisione. Questo è il titolo che viene visualizzato nell'elenco di richieste pull per il repository. In Description (Descrizione), immettere i dettagli sul contenuto della revisione ed eventuali altre informazioni utili per i revisori. Scegli Create (Crea) .

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: I've added some code for the bucket creation issue. Please review by Tuesday.
☐ Preview markdown [Learn more](#)

Changes Commits

4. La richiesta di pull viene visualizzata nell'elenco di richieste pull per il repository. È possibile applicare un filtro alla visualizzazione perché siano visibili solo le richieste aperte, le richieste chiuse, le richieste create dall'utente e altro ancora.

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. Puoi aggiungere una regola di approvazione alla tua pull request per assicurarti che vengano soddisfatte determinate condizioni prima che possa essere unita. Per aggiungere una regola di approvazione alla richiesta pull, scegliere la richiesta pull dall'elenco. Nella scheda Approvals (Approvazioni), scegliere Create approval rule (Crea regola di approvazione).

6. In Rule name (Nome regola), assegnare alla regola un nome descrittivo. Ad esempio, se si desidera che due persone approvino una richiesta pull prima che possa essere unita, è possibile assegnare un nome alla regola **Require two approvals before merge**. In Number of approvals needed (Numero di approvazioni necessarie), immettere **2**, il numero desiderato. Il valore di default è 1. Scegli Invia. Per ulteriori informazioni sulle regole di approvazione e sui membri del pool di approvazione, vedere [Creare una regola di approvazione per una pull request](#).

Create approval rule

Rule details

Rule name
Require two approvals before merge

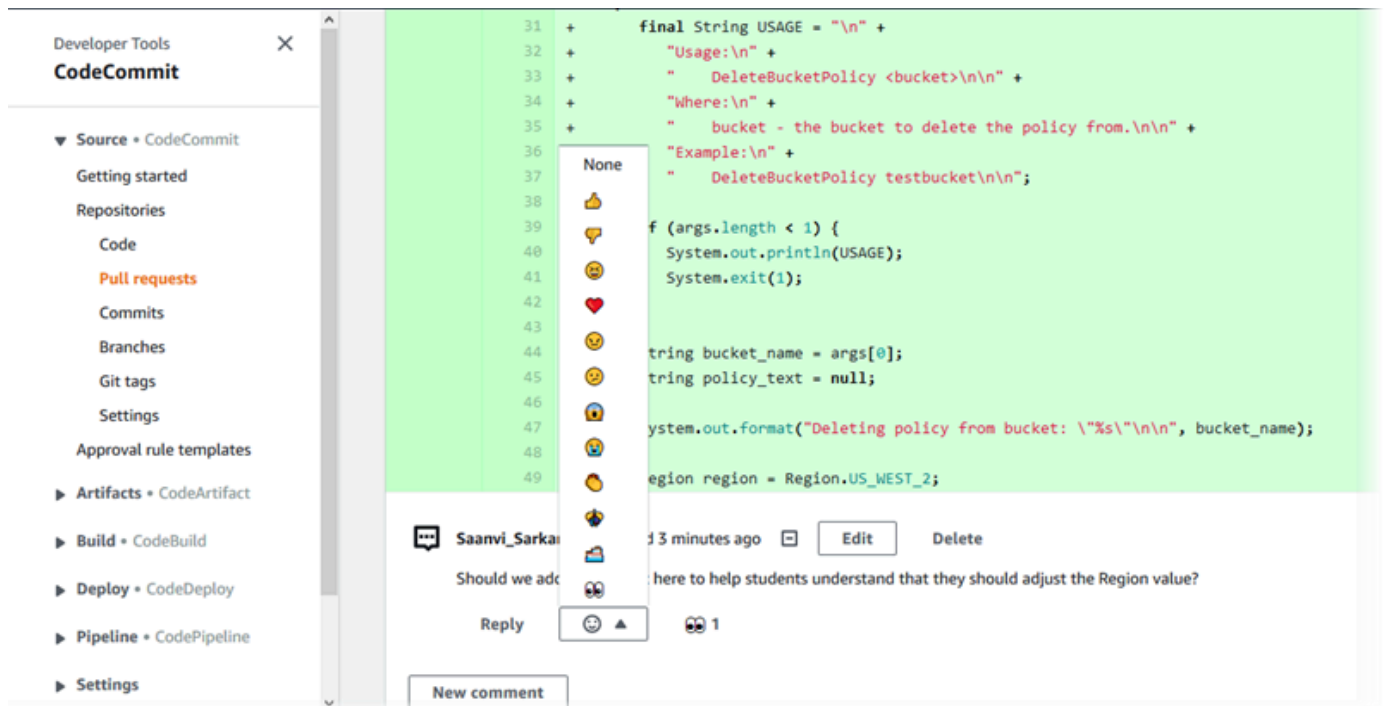
Number of approvals needed
2

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.

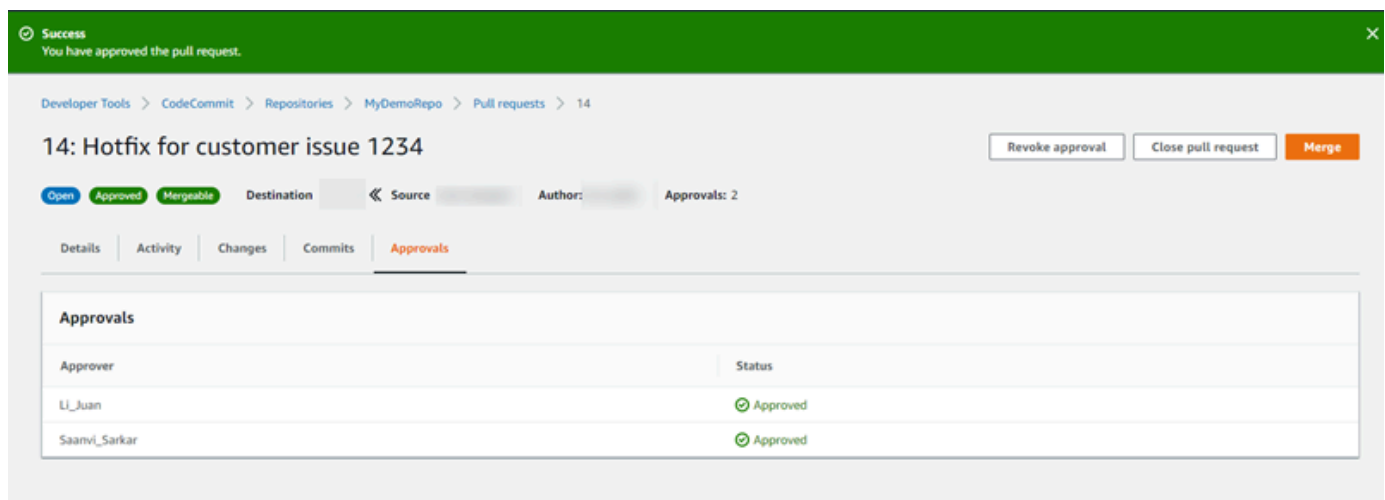
Add

Cancel Submit

7. Se sono state configurate le notifiche per il repository e si è scelto di notificare agli utenti gli eventi di richiesta di pull, gli utenti riceveranno un'e-mail relativa alla nuova richiesta di pull. Gli utenti possono visualizzare le modifiche e commentare righe di codice specifiche, file e la stessa richiesta di pull. Possono anche rispondere ai commenti con testo ed emoji. Se necessario, è possibile inviare le modifiche al ramo di richiesta di pull, che aggiorna quest'ultima.



8. Se sei soddisfatto delle modifiche apportate nella richiesta, scegli Approve Approva. È possibile scegliere di approvare una richiesta pull anche se non sono configurate regole di approvazione per tale richiesta pull. Ciò rappresenta una chiara conferma dell'esame della richiesta di pull e dell'approvazione delle modifiche. Puoi anche scegliere di revocare la tua approvazione se cambi idea.



Note

Non è possibile approvare una richiesta pull se è stata creata.

9. Quando tutte le modifiche al codice sono stati riviste e accettate, dalla richiesta di pull procedere in uno dei seguenti modi:
- Se si desidera chiudere la richiesta di pull senza unire i rami, scegliere Close pull request (Chiudi pull di richiesta).
 - Se si desidera unire i rami e chiudere la richiesta pull, scegliere Unisci. È possibile scegliere tra le strategie di unione disponibili per il codice, che dipendono dalle differenze tra i rami di origine e di destinazione e se eliminare automaticamente il ramo di origine una volta completata l'unione. Una volta effettuate le proprie scelte, selezionare Merge pull request (Unisci richiesta di pull) per completare l'unione.

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

Email address

☒ Delete source branch bugfix-bug1234 after merging?

Cancel Merge pull request

- Se ci sono conflitti di unione nei rami che non possono essere risolti automaticamente, puoi risolverli nella CodeCommit console oppure puoi usare il tuo client Git locale per unire i rami e quindi eseguire l'unione. Per ulteriori informazioni, consulta [Risolvi i conflitti in una pull request in un AWS CodeCommit repository](#).

 Note

È sempre possibile un'unione manuale dei rami, compresi i rami della richiesta di pull, utilizzando il comando `git merge` nel repository locale e applicando le proprie modifiche.

Per ulteriori informazioni, consulta [Utilizzo delle richieste pull](#) e [Utilizzo dei modelli di regola di approvazione](#).

Fase 5: rimozione

Se il CodeCommit repository non è più necessario, è consigliabile eliminare il CodeCommit repository e le altre risorse utilizzate in questo esercizio in modo da non continuare a pagare per lo spazio di archiviazione.

 Important

Questa operazione non può essere annullata. Dopo aver eliminato questo repository, non è più possibile clonarlo su alcun repository locale o condiviso. Inoltre, non puoi più estrarre dati da o inviare dati ad esso, o eseguire operazioni Git, da qualsiasi repository locale o repository condiviso.

Se hai configurato le notifiche per il tuo repository, l'eliminazione del repository elimina anche la regola CloudWatch Amazon Events creata per il repository. Non elimina l'argomento Amazon SNS utilizzato come destinazione per quella regola.

Se hai configurato i trigger per il tuo repository, l'eliminazione del repository non elimina gli argomenti di Amazon SNS o le funzioni Lambda che hai configurato come destinazioni di tali trigger. Assicurati di eliminare tali risorse se non ti sono necessarie. Per ulteriori informazioni, consulta [Eliminare i trigger da un repository](#).

CodeCommit Per eliminare il repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository) selezionare il repository da eliminare. Se è stata seguita la convenzione illustrata in questo argomento, è denominata MyDemoRepo.
3. Nel pannello di navigazione scegli Impostazioni.

4. Nella pagina Settings (Impostazioni) in Delete repository (Elimina repository), scegliere Delete repository (Elimina repository).
5. Digitare **delete** e scegliere Delete (Elimina). Il repository viene eliminato definitivamente.

Fase 6: fasi successive

Ora che hai acquisito familiarità con CodeCommit alcune delle sue funzionalità, prova a fare quanto segue:

- Se non conosci Git CodeCommit o vuoi rivedere esempi di utilizzo di Git with CodeCommit, continua con il [Guida introduttiva a Git e CodeCommit](#) tutorial.
- Se vuoi lavorare con altri utenti in un CodeCommit repository, consulta [Condividi un repository](#). (Se desideri condividere il tuo repository con gli utenti di un altro account Amazon Web Services, consulta [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#).)
- Se desideri migrare un repository in CodeCommit, segui i passaggi indicati in. [Migrare a CodeCommit](#)
- Se vuoi aggiungere il tuo repository a una pipeline di distribuzione continua, procedi come descritto nella [procedura guidata per una pipeline semplice](#).
- Se vuoi ulteriori informazioni su prodotti e servizi che si integrano con CodeCommit, inclusi esempi della community, consulta [Integrazioni di prodotti e servizi](#).

Guida introduttiva a Git e AWS CodeCommit

Se non conosci Git e CodeCommit, questo tutorial ti aiuta a imparare alcuni semplici comandi per iniziare. Se conosci già Git, puoi ignorare questo tutorial e passare al [Iniziare con CodeCommit](#).

In questo tutorial, crei un repository che rappresenta una copia locale del CodeCommit repository, che chiamiamo repository locale.

Dopo aver creato il repository locale, apporti alcune modifiche ad esso. Quindi invii (invia) le modifiche al CodeCommit repository.

Inoltre, simuli un ambiente di gruppo in cui due utenti eseguono in modo indipendente le modifiche al loro repository locale e inviano tali modifiche al repository. CodeCommit Gli utenti trasferiscono quindi le modifiche dal repository al proprio CodeCommit repository locale per vedere le modifiche apportate dall'altro utente.

Puoi anche creare rami e tag e gestire alcune autorizzazioni di accesso nel repository. CodeCommit

Al termine di questo tutorial, dovresti aver acquisito una conoscenza sufficiente dei concetti di CodeCommit per utilizzarli per i tuoi progetti.

Esegui le [operazioni relative ai prerequisiti e alla configurazione](#) tra cui:

- Assegna le autorizzazioni all'utente IAM.
- Configura CodeCommit la connessione a un repository tramite [HTTPS](#), SSH o [git-remote-codecommit](#). Per ulteriori informazioni su queste scelte, consulta [Configurazione per AWS CodeCommit](#).
- Configura AWS CLI se desideri utilizzare la riga di comando o il terminale per tutte le operazioni, inclusa la creazione del repository.

Argomenti

- [Fase 1: Creare un repository CodeCommit](#)
- [Fase 2: Creare un repository locale](#)
- [Passaggio 3: crea il tuo primo commit](#)
- [Passaggio 4: invia il tuo primo commit](#)
- [Passaggio 5: condividi il CodeCommit repository e invia ed estrai un altro commit](#)
- [Passaggio 6: Crea e condividi un ramo](#)
- [Passaggio 7: Crea e condividi un tag](#)
- [Passaggio 8: configurare le autorizzazioni di accesso](#)
- [Fase 9: Pulizia](#)

Fase 1: Creare un repository CodeCommit

In questo passaggio, si utilizza la CodeCommit console per creare il repository.

Puoi saltare questo passaggio se hai già un CodeCommit repository che desideri utilizzare.

 Note

A seconda dell'utilizzo, potrebbero esserti addebitati costi per la creazione o l'accesso a un repository. Per ulteriori informazioni, consulta la sezione [Prezzi](#) nella pagina di informazioni sul CodeCommit prodotto.

Per creare il CodeCommit repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Usa il selettore della regione per scegliere Regione AWS dove vuoi creare il repository. Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository), scegli Create repository (Crea repository).
4. Nella pagina Create repository (Crea repository), in Repository name (Nome repository), immetti un nome per il repository (ad esempio **MyDemoRepo**).

 Note

I nomi dei repository distinguono tra lettere maiuscole e minuscole e non possono superare i 100 caratteri. Per ulteriori informazioni, consulta [Limiti](#).

5. (Facoltativo) In Description (Descrizione), immettere una descrizione, ad esempio **My demonstration repository**. Ciò può consentire agli utenti di identificare lo scopo del repository.
6. (Facoltativo) Scegliete Aggiungi tag per aggiungere uno o più tag del repository (un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse) al repository. Per ulteriori informazioni, consulta [Taggare i repository in AWS CodeCommit](#).
7. (Facoltativo) Espandi la configurazione aggiuntiva per specificare se utilizzare la chiave predefinita Chiave gestita da AWS o la chiave gestita dal cliente per crittografare e decrittografare i dati in questo repository. Se scegli di utilizzare la tua chiave gestita dal cliente, devi assicurarti che sia disponibile nel luogo in Regione AWS cui stai creando il repository e che la chiave sia attiva. Per ulteriori informazioni, consulta [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#).
8. (Facoltativo) Seleziona Abilita Amazon CodeGuru Reviewer per Java e Python se questo repository conterrà codice Java o Python e desideri che Reviewer analizzi CodeGuru quel codice. CodeGuru Reviewer utilizza diversi modelli di machine learning per trovare i difetti del

codice e suggerire automaticamente miglioramenti e correzioni nelle pull request. Per ulteriori informazioni, consulta la Amazon CodeGuru Reviewer User Guide.

9. Scegli Create (Crea).

Note

I passaggi rimanenti di questo tutorial vengono utilizzati MyDemoRepo per il nome del CodeCommit repository. Se scegli un nome differente, assicurati di utilizzarlo in tutto il tutorial.

Per ulteriori informazioni sulla creazione di repository, tra cui la creazione di un repository dal terminale o dalla riga di comando, consulta [Creazione di un repository](#).

Fase 2: Creare un repository locale

In questa fase, puoi configurare un repository locale sul computer locale per la connessione al tuo repository. A questo proposito, selezioni una directory sul computer locale che rappresenta il repository locale. Usi Git per clonare e inizializzare una copia del tuo CodeCommit repository vuoto all'interno di quella directory. Quindi specifichi il nome utente e l'indirizzo email Git usati per annotare i tuoi commit.

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Trova e seleziona nell'elenco il repository che intendi collegare. Scegliere Clone URL (URL clone), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione o la connessione al repository. In questo modo si copia l'URL del clone.
 - Copia l'URL HTTPS se utilizzi le credenziali Git con il tuo utente IAM o l'helper per le credenziali incluso in. AWS CLI
 - Copia l'URL HTTPS (GRC) se usi il comando git-remote-codecommit sul computer locale.
 - Copia l'URL SSH se utilizzi una coppia di public/private chiavi SSH con il tuo utente IAM.

 Note

Se vedi una pagina di benvenuto anziché un elenco di repository, significa che non ci sono repository associati al tuo AWS account nel luogo in Regione AWS cui hai effettuato l'accesso. Per creare un repository, consulta [the section called “Creazione di un repository”](#) oppure segui la procedura del tutorial [Guida introduttiva a Git e CodeCommit](#).

4. (Facoltativo) Ti consigliamo di configurare il tuo client Git locale da utilizzare **main** come nome per il ramo predefinito del tuo repository. Questo è il nome usato per il ramo predefinito in tutti gli esempi di questa guida. È anche lo stesso nome di ramo predefinito CodeCommit utilizzato per il primo commit nella console. Esegui il seguente comando per configurare globalmente il nome del ramo predefinito per il tuo sistema:

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

Se preferisci utilizzare un nome di ramo predefinito diverso per tutti i tuoi repository, sostituiscilo **main** con il tuo nome preferito. Questo tutorial presuppone che il ramo predefinito sia denominato main.

Se desideri utilizzare nomi di ramo predefiniti diversi per repository diversi, puoi impostare questo attributo localmente (--local) anziché globalmente (--global).

5. Al terminale o al prompt dei comandi, clonate il repository con il git clone comando e fornite l'URL del clone copiato nel passaggio 3. L'URL clone dipende dal protocollo e dalla configurazione utilizzati. Ad esempio, se utilizzi HTTPS con credenziali Git per clonare un repository denominato *MyDemoRepo* nella regione Stati Uniti orientali (Ohio):

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

Se stai usando HTTPS con git-remote-codecommit:

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

Se stai usando SSH:

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

Note

Se viene visualizzato un errore durante il tentativo di clonare un repository, è possibile che non sia stata completata la configurazione necessaria per il computer locale. Per ulteriori informazioni, consulta [Configurazione per AWS CodeCommit](#).

Passaggio 3: crea il tuo primo commit

In questo passaggio, crei il tuo primo commit nel tuo repository locale. Per fare ciò, crei due file di esempio nel repository locale. Si utilizza Git per effettuare la modifica e quindi eseguire il commit della modifica nel repository locale.

1. Utilizza un editor di testo per creare i seguenti due file di testo di esempio nella directory. Denomina i file `cat.txt` e `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. Esegui `git config` per aggiungere il nome utente e l'indirizzo email rappresentati dai segnaposto *your-user-name* e *your-email-address* al tuo repository locale. Ciò rende più semplice identificare i commit eseguiti:

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

3. Se non hai impostato il nome del ramo predefinito a livello globale quando hai creato il repository locale, esegui il comando seguente per impostare il nome del ramo predefinito su: **main**

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

4. Esegui `git add` per preparare la modifica:

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

5. Esegui `git commit` per eseguire il commit della modifica:

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

Tip

Per visualizzare i dettagli sul commit, esegui `git log`.

Passaggio 4: invia il tuo primo commit

In questo passaggio, trasferisci il commit dal repository locale al tuo CodeCommit repository.

Esegui `git push` per inviare il tuo commit tramite il nome remoto predefinito che Git usa per il tuo CodeCommit repository (`origin`), dal ramo predefinito nel tuo repository locale (`main`):

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

Tip

Dopo aver inviato i file al CodeCommit repository, puoi usare la CodeCommit console per visualizzarne i contenuti. Per ulteriori informazioni, consulta [Sfoggia i file in un repository](#).

Passaggio 5: condividi il CodeCommit repository e invia ed estrai un altro commit

In questo passaggio, condividi le informazioni sul CodeCommit repository con un altro membro del team. Il membro del team utilizza queste informazioni per ottenere una copia locale, apportare alcune

modifiche e quindi inviare la copia locale modificata al CodeCommit repository. Quindi trasferisci le modifiche dal repository al tuo CodeCommit repository locale.

In questo tutorial, simuli l'altro utente creando una directory distinta da quello creata nella [fase 2](#) utilizzando Git. In genere, questa directory si trova su un altro computer. Questa nuova directory è una copia del tuo CodeCommit repository. Qualsiasi modifica apportata alla directory esistente o a questa nuova directory viene eseguita in modo indipendente. L'unico modo per identificare le modifiche a queste directory è scaricarle dal CodeCommit repository.

Anche se queste directory si trovano sulla stessa macchina locale, denominiamo quella esistente repository locale e la nuova directory repository condiviso.

Dalla nuova directory, si ottiene una copia separata del CodeCommit repository. Quindi aggiungi un nuovo file di esempio, esegui il commit delle modifiche nel repository condiviso e quindi trasferisci il commit dal repository condiviso al tuo repository. CodeCommit

Infine, trasferisci le modifiche dal tuo repository al repository locale e poi lo sfogli per vedere le modifiche apportate dall'altro utente.

1. Passa alla directory `/tmp` o alla directory `c:\temp`.
2. Esegui `git clone` per scaricare una copia del repository nel repository condiviso:

Per HTTPS:

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

Per HTTPS con `git-remote-codecommit`:

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

Per SSH:

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

 Note

Quando crei un clone di un repository utilizzando SSH nei sistemi operativi Windows, potrebbe essere necessario aggiungere l'ID della chiave SSH alla stringa di connessione come segue:

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

Per ulteriori informazioni, consulta [Per le connessioni SSH su Windows](#).

In questo comando, MyDemoRepo c'è il nome del tuo repository. CodeCommit shared-demo-repo è il nome della directory che Git crea nella /tmp directory o nella c:\temp directory. Dopo che Git ha creato la directory, scarica una copia del repository nella directory shared-demo-repo.

3. Passa alla directory shared-demo-repo:

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

4. Esegui git config per aggiungere un altro nome utente e indirizzo email rappresentati da segnaposto *other-user-name* e *other-email-address*. Ciò rende più semplice identificare i commit eseguiti dell'altro utente:

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

5. Utilizza un editor di testo per creare i seguenti file di testo di esempio nella directory shared-demo-repo. Denomina il file horse.txt:

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

6. Esegui git add per preparare la modifica al repository condiviso:

```
git add horse.txt
```

7. Esegui `git commit` per effettuare il commit della modifica nel repository condiviso:

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

8. Esegui `git push` per inviare il tuo commit iniziale tramite il nome remoto predefinito che Git usa per il tuo CodeCommit repository (`origin`), dal ramo predefinito nel tuo repository locale (`main`):

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

9. Passa al repository locale ed esegui `git pull` per inserire nel repository locale il commit che il repository condiviso ha effettuato sul repository. CodeCommit Esegui quindi `git log` per vedere il commit avviato dal repository condiviso.

Passaggio 6: Crea e condividi un ramo

In questo passaggio, crei un ramo nel repository locale, apporti alcune modifiche e quindi trasferisci il ramo al tuo CodeCommit repository. Quindi trasferisci il ramo al repository condiviso dal tuo repository. CodeCommit

Un ramo consente di sviluppare in modo indipendente una versione differente del contenuto del repository (ad esempio, per lavorare su una nuova funzionalità software senza influenzare il lavoro dei membri del team). Quando quella funzionalità è stabile, unisci il ramo a un ramo più stabile del software.

Utilizzi Git per creare il ramo e quindi puntarlo al primo commit effettuato. Si usa Git per inviare il ramo al CodeCommit repository. Quindi passi al repository condiviso e usi Git per inserire il nuovo ramo nel repository locale condiviso ed esplorare il ramo.

1. Dal repository locale, esegui `git checkout`, specificando il nome del ramo (ad esempio `MyNewBranch`) e l'ID del primo commit che hai effettuato nel repository locale.

Se non conosci l'ID commit, esegui `git log` per ottenerlo. Verifica che il commit abbia il tuo nome utente e il tuo indirizzo e-mail e non il nome utente e l'indirizzo e-mail dell'altro utente. Questo serve a simulare che `main` si tratti di una versione stabile del CodeCommit repository e che il `MyNewBranch` branch riguardi una nuova funzionalità relativamente instabile:

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

2. Esegui `git push` per inviare il nuovo ramo dal repository locale al repository: CodeCommit

```
git push origin MyNewBranch
```

3. A questo punto, esegui il pull del ramo al repository condiviso e controlla i risultati:
 1. Passa alla directory del repository condiviso (shared-demo-repo).
 2. Esegui il pull nel nuovo ramo (git fetch origin).
 3. Verifica che il pull del ramo sia stato eseguito (git branch --all visualizza un elenco di tutti rami per il repository).
 4. Passa al nuovo ramo (git checkout MyNewBranch).
 5. Verifica l'avvenuto passaggio al ramo MyNewBranch eseguendo git status o git branch. L'output mostra il ramo corrente. In questo caso, deve essere MyNewBranch.
 6. Visualizza l'elenco dei commit nel ramo (git log).

Di seguito è riportato l'elenco dei comandi Git da chiamare:

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

4. Passa di nuovo al ramo main e visualizzane l'elenco di commit. I comandi Git devono risultare simili a quanto segue:

```
git checkout main
git log
```

5. Passa alla main filiale del tuo repository locale. Puoi eseguire git status o git branch. L'output mostra il ramo corrente. In questo caso, deve essere main. I comandi Git devono risultare simili a quanto segue:

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

Passaggio 7: Crea e condividi un tag

In questo passaggio, crei due tag nel tuo repository locale, associ i tag ai commit e quindi inserisci i tag nel tuo CodeCommit repository. Quindi trasferisci le modifiche dal CodeCommit repository al repository condiviso.

Un tag è utilizzato per fornire un nome leggibile a un commit (o a un ramo o a un altro tag). Lo faresti, ad esempio, per taggare un commit come v2.1. Un commit, ramo o tag può avere un qualsiasi numero di tag a esso associato, ma un singolo tag può essere associato a un solo commit, ramo o tag. In questo tutorial, applichi un tag a un commit come `release` e uno come `beta`.

Utilizzi Git per creare i tag, puntando il tag `release` al primo commit eseguito e il tag `beta` al commit eseguito dall'altro utente. Quindi usi Git per inviare i tag al CodeCommit repository. Quindi passi al tuo repository condiviso e usi Git per inserire i tag nel tuo repository locale condiviso ed esplorare i tag.

1. Dal tuo repository locale, esegui `git tag`, specificando il nome del nuovo tag (`release`) e l'ID del primo commit che hai effettuato nel repository locale.

Se non conosci l'ID commit, esegui `git log` per ottenerlo. Verifica che il commit abbia il tuo nome utente e il tuo indirizzo e-mail e non il nome utente e l'indirizzo e-mail dell'altro utente. Questo serve a simulare che il tuo commit sia una versione stabile del repository: CodeCommit

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

Esegui di nuovo `git tag` per taggare il commit di un altro utente con il tag `beta`. Ciò è necessario per simulare che il commit è destinato a una nuova funzionalità relativamente instabile:

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

2. Esegui `git push --tags` per inviare i tag al repository. CodeCommit
3. A questo punto, esegui il pull dei tag nel repository condiviso e controlla i risultati:
 1. Passa alla directory del repository condiviso (`shared-demo-repo`).
 2. Esegui il pull dei nuovi tag (`git fetch origin`).
 3. Verifica che il pull dei tag è stato eseguito (`git tag` visualizza un elenco di tutti i tag per il repository).
 4. Visualizza le informazioni su ogni tag (`git log release` e `git log beta`).

Di seguito è riportato l'elenco dei comandi Git da chiamare:

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

4. Provalo anche nel repository locale:

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

Passaggio 8: configurare le autorizzazioni di accesso

In questo passaggio, concedi all'utente l'autorizzazione a sincronizzare il repository condiviso con il repository. CodeCommit Questa fase è opzionale. È consigliato agli utenti interessati a scoprire come controllare l'accesso ai CodeCommit repository quando gli utenti utilizzano credenziali Git o coppie di chiavi SSH vengono utilizzate con gli utenti IAM per l'accesso ai repository. CodeCommit

Note

Se utilizzi l'accesso federato, credenziali temporanee o un provider di identità web come IAM Identity Center, configura gli utenti, l'accesso e le autorizzazioni per il tuo provider di identità, quindi utilizza `git-remote-codecommit`. Per ulteriori informazioni, consultare [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#) e [Connessione ai AWS CodeCommit repository con credenziali rotanti](#).

A tale scopo, utilizzi la console IAM per creare un utente che, per impostazione predefinita, non dispone delle autorizzazioni per sincronizzare il repository condiviso con il repository. CodeCommit Puoi eseguire `git pull` per verificarlo. Se il nuovo utente non dispone delle autorizzazioni per eseguire la sincronizzazione, il comando non funziona. Quindi torni alla console IAM e applichi una policy che ne consenta l'utilizzo da parte dell'utente. `git pull` Anche in questo caso, puoi eseguire `git pull` per verificarlo.

Questo passaggio è scritto presupponendo che tu disponga delle autorizzazioni per creare utenti IAM nel tuo account Amazon Web Services. Se non disponi di tali autorizzazioni, non puoi eseguire le procedure in questa fase. Passa alla [Fase 9: Pulizia](#) per pulire le risorse utilizzate per il tutorial.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>

Assicurati di effettuare l'accesso con il nome utente e la password utilizzati in [Configurazione](#).

2. Nel riquadro di navigazione, seleziona Users (Utenti), quindi scegli Create New Users (Crea nuovi utenti).
3. Nella prima casella Enter User Names (Immetti i nomi utente), immetti un nome utente di esempio (ad esempio, **JaneDoe-CodeCommit**). Seleziona la casella Generate an access key for each user (Genera una chiave di accesso per ogni utente), quindi scegli Create (Crea).
4. Scegli Show User SMTP Credentials (Mostra credenziali di sicurezza utente). Annota l'ID chiave di accesso e la chiave di accesso segreta oppure scegli Download Credentials (Credenziali di download).
5. Segui le istruzioni [Per utenti HTTPS che utilizzano credenziali Git](#) per generare e fornire le credenziali dell'utente IAM.

Se desideri utilizzare SSH, segui le istruzioni in [SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit](#) o [Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit](#) per configurare l'utente con le chiavi private e pubbliche.

6. Esegui git pull. Deve essere visualizzato l'errore seguente:

Per HTTPS:

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

Per SSH:

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

L'errore viene visualizzato perché il nuovo utente non dispone dell'autorizzazione per sincronizzare il repository condiviso con il repository. CodeCommit

7. Torna alla console IAM. Nel riquadro di navigazione, seleziona Policies (Policy) e Create Policy (Crea policy). (Se viene visualizzato il pulsante Get Started (Inizia), sceglierlo, quindi scegliere Create Policy (Crea policy)).
8. Accanto a Create Your Own Policy (Crea la tua policy) scegli Select (Seleziona).
9. Nella casella Policy Name (Nome policy), immetti un nome (ad esempio **CodeCommitAccess-GettingStarted**).
10. Nella casella Policy Document, inserisci quanto segue, che consente a un utente IAM di estrarre da qualsiasi repository associato all'utente IAM:

JSON

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

 Tip

Se desideri che l'utente IAM sia in grado di inviare i commit a qualsiasi repository associato all'utente IAM, inserisci invece questo:

JSON

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



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

Per informazioni su altre autorizzazioni per CodeCommit azioni e risorse che puoi concedere agli utenti, consulta. [Autenticazione e controllo degli accessi per AWS CodeCommit](#)

11. Nel pannello di navigazione, seleziona Utenti.
12. Scegli il nome utente di esempio (ad esempio, **JaneDoe-CodeCommit**) a cui desideri collegare la policy.
13. Scegli la scheda Autorizzazioni.
14. In Managed Policies (Policy gestite), scegli Attach Policy (Collega policy).
15. Seleziona la policy **CodeCommitAccess-GettingStarted** appena creata, quindi scegli Attach policy (Collega policy).
16. Esegui git pull. Questa volta il comando dovrebbe funzionare e dovrebbe essere visualizzato un messaggio Already up-to-date.
17. Se stai usando HTTPS, passa alle credenziali Git originali oppure se stai utilizzando git-remote-codecommit, al tuo profilo abituale. Per ulteriori informazioni, consulta le istruzioni in [Configurazione per utenti HTTPS che utilizzano credenziali Git](#) o [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Se utilizzi SSH, passa alle chiavi originali. Per ulteriori informazioni, consulta [SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit](#) o [Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit](#).

Il tutorial è terminato.

Fase 9: Pulizia

In questo passaggio, elimini l' CodeCommit archivio che hai usato in questo tutorial, in modo da non continuare a ricevere addebiti per lo spazio di archiviazione.

Rimuoverai anche il repository locale e il repository condiviso sul tuo computer locale perché non saranno necessari dopo aver eliminato il repository. CodeCommit

Important

Dopo aver eliminato questo repository, non sarà possibile clonarlo su alcun repository locale o condiviso. Inoltre, non sarai in grado di estrarre dati da esso o inviargli dati da alcun repository locale o condiviso. Questa operazione non può essere annullata.

Per eliminare il CodeCommit repository (console)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nella pagina Dashboard (Pannello di controllo), nell'elenco di repository, scegli MyDemoRepo.
3. Nel pannello di navigazione scegli Impostazioni.
4. Nella pagina Settings (Impostazioni) in Delete repository (Elimina repository), scegliere Delete repository (Elimina repository).
5. Nella casella accanto a Type the name of the repository to confirm deletion (Digita il nome del repository per confermare l'eliminazione), immettere **MyDemoRepo** e quindi scegliere Delete (Elimina).

Per eliminare il repository () CodeCommit AWS CLI

Esegui il comando [delete-repository](#):

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

Per eliminare il repository locale e il repository condiviso

Per Linux, macOS o Unix:

```
cd /tmp
```

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

Per Windows:

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

Integrazioni di prodotti e servizi con AWS CodeCommit

Per impostazione predefinita, CodeCommit è integrato con una serie di AWS servizi. È inoltre possibile utilizzarlo CodeCommit con prodotti e servizi esterni a AWS. Le informazioni seguenti possono essere utili per configurare CodeCommit per l'integrazione con i prodotti e i servizi utilizzati.

Argomenti

- [Integrazione con altri AWS servizi](#)
- [Esempi di integrazione dalla community](#)

Integrazione con altri AWS servizi

CodeCommit è integrato con i seguenti AWS servizi:

AWS Amplify

[AWS Amplify](#) semplifica la creazione, la configurazione e l'implementazione di applicazioni mobili scalabili basate su AWS. Amplify esegue il provisioning e gestisce il back-end per dispositivi mobili in modo ottimale, oltre a offrire un framework semplice per integrare il back-end con i front-end su iOS, Android, Web e React Native. Amplify automatizza anche il processo di rilascio dell'applicazione sia per il front-end, sia per il back-end, in modo da consentire di distribuire le caratteristiche più rapidamente.

Puoi connettere il tuo CodeCommit repository nella console Amplify. Dopo aver autorizzato la console Amplify, Amplify recupera un token di accesso dal provider del repository, ma non lo memorizza sui server. AWS Amplify accede al repository utilizzando chiavi di distribuzione installate solo in uno specifico repository.

Ulteriori informazioni:

- [AWS Amplify Guida per l'utente](#)
- [Nozioni di base](#)

AWS Cloud9

[AWS Cloud9](#) contiene una raccolta di strumenti utilizzati per programmare, creare, eseguire, testare, eseguire il debug e rilasciare software nel cloud. Questa raccolta di strumenti è denominata ambiente di sviluppo integrato o IDE. AWS Cloud9

Si accede all' AWS Cloud9 IDE tramite un browser web. L'IDE offre una ricca esperienza di modifica del codice con supporto per diversi linguaggi di programmazione e debugger nel tempo di esecuzione, nonché un terminale integrato.

Ulteriori informazioni:

- [AWS Cloud9 Guida per l'utente](#)
- [AWS CodeCommit Esempio per AWS Cloud9](#)
- [Integrazione AWS Cloud9 con AWS CodeCommit](#)

CloudFormation

[CloudFormation](#) è un servizio che consente di modellare e configurare le AWS risorse in modo da dedicare meno tempo alla gestione di tali risorse e più tempo alle applicazioni. Crei un modello che descrive le risorse, incluso un CodeCommit repository, e CloudFormation si occupa del provisioning e della configurazione di tali risorse per te.

Ulteriori informazioni:

- [AWS CodeCommit Pagina delle risorse del repository](#)

AWS CloudTrail

[CloudTrail](#) acquisisce le chiamate AWS API e gli eventi correlati effettuati da o per conto di un account Amazon Web Services e invia i file di registro a un bucket Amazon S3 specificato dall'utente. È possibile CloudTrail configurare l'acquisizione delle chiamate API dalla AWS CodeCommit console AWS CLI, dei CodeCommit comandi dal client Git locale e dall' CodeCommit API.

Ulteriori informazioni:

- [Registrazione delle chiamate AWS CodeCommit API con AWS CloudTrail](#)

CloudWatch Eventi Amazon

[CloudWatch Events](#) offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. Utilizzando regole semplici che puoi configurare rapidamente, puoi abbinare gli eventi e indirizzarli a una o più funzioni o flussi di destinazione. CloudWatch Gli eventi vengono a conoscenza dei cambiamenti operativi man mano che si verificano. CloudWatch Events risponde a questi cambiamenti operativi e interviene se necessario, inviando messaggi per rispondere e all'ambiente, attivando funzioni, apportando modifiche e acquisendo informazioni sullo stato.

Puoi configurare CloudWatch Events per monitorare i CodeCommit repository e rispondere agli eventi dei repository indirizzando flussi, funzioni, attività o altri processi in altri AWS servizi, come Amazon Simple Queue Service, Amazon Kinesis e molti altri. AWS Lambda

Ulteriori informazioni:

- [CloudWatch Guida per l'utente degli eventi](#)
- [Eventi AWS CodeCommit](#)
- Post del blog: [Crea AWS CodeCommit flussi di lavoro serverless utilizzando Amazon CloudWatch Events](#) e JGit

AWS CodeBuild

[CodeBuild](#) è un servizio di compilazione completamente gestito nel cloud che permette di compilare il tuo codice sorgente, eseguire test e produrre elementi pronti per essere distribuiti. Puoi archiviare il codice sorgente da creare e la specifica della compilazione in un repository CodeCommit. Puoi utilizzarli CodeBuild direttamente con CodeCommit, oppure puoi incorporarli entrambi CodeBuild e CodeCommit in una pipeline di distribuzione continua con CodePipeline

Ulteriori informazioni:

- [Pianifica una compilazione](#)
- [Crea un Progetto di compilazione](#)
- [Usalo CodePipeline con AWS CodeBuild per eseguire build](#)

CodeGuru Revisore Amazon

Amazon CodeGuru Reviewer è un servizio automatizzato di revisione del codice che utilizza l'analisi dei programmi e l'apprendimento automatico per rilevare problemi comuni e consigliare correzioni nel codice Java o Python. Puoi associare i repository nel tuo account Amazon Web Services a CodeGuru Reviewer. Quando lo fai, CodeGuru Reviewer crea un ruolo collegato al servizio che consente a CodeGuru Reviewer di analizzare il codice in tutte le pull request create dopo l'associazione. Ulteriori informazioni:

- [Associare o dissociare un AWS CodeCommit repository con Amazon Reviewer CodeGuru](#)
- [Guida per l'utente CodeGuru di Amazon Reviewer](#)

AWS CodePipeline

[CodePipeline](#) è un servizio di distribuzione continua che può essere utilizzato per modellare, visualizzare e automatizzare le fasi necessarie al rilascio di software. Puoi configurare l'utilizzo CodePipeline di un CodeCommit repository come azione di origine in una pipeline e automatizzare la creazione, il test e la distribuzione delle modifiche.

Ulteriori informazioni:

- [Semplice procedura dettagliata sulla pipeline con e CodePipeline AWS CodeCommit](#)
- [Esegui la migrazione ad Amazon CloudWatch Events Change Detection for Pipeline con un repository CodeCommit](#)
- [Metodi di rilevamento delle modifiche utilizzati per avviare automaticamente le pipeline](#)

AWS Elastic Beanstalk

[Elastic Beanstalk](#) è un servizio gestito che semplifica l'implementazione e la gestione delle applicazioni AWS nel cloud senza preoccuparsi dell'infrastruttura che esegue tali applicazioni. Puoi utilizzare l'interfaccia a riga di comando di Elastic Beanstalk (EB CLI) per distribuire l'applicazione direttamente da un repository nuovo o esistente. CodeCommit

Ulteriori informazioni:

- [Utilizzo della CLI EB con AWS CodeCommit](#)
- [Utilizzo AWS CodeCommit di un repository esistente](#)
- [eb codesource \(comando CLI EB\)](#)

AWS Key Management Service

[AWS KMS](#) è un servizio gestito che semplifica la creazione e il controllo di chiavi di crittografia per la codifica dei dati. Per impostazione predefinita, viene utilizzato AWS KMS per crittografare i repository.

Ulteriori informazioni:

- [AWS KMS e crittografia](#)

AWS Lambda

[Lambda](#) consente di eseguire codice senza effettuare il provisioning o gestire i server. È possibile configurare i trigger per i CodeCommit repository che richiamano le funzioni Lambda in risposta agli eventi del repository.

Ulteriori informazioni:

- [Creare un trigger per una funzione Lambda](#)
- [AWS Lambda Guida per gli sviluppatori](#)

Amazon Simple Notification Service

[Amazon SNS](#) è un servizio Web che consente alle applicazioni, agli utenti finali e ai dispositivi di inviare e ricevere istantaneamente notifiche dal cloud. Puoi configurare i trigger per i CodeCommit repository che inviano notifiche Amazon SNS in risposta agli eventi del repository. Puoi anche utilizzare le notifiche di Amazon SNS per l'integrazione con altri AWS servizi. Ad esempio, puoi utilizzare una notifica Amazon SNS per inviare messaggi a una coda di Amazon Simple Queue Service.

Ulteriori informazioni:

- [Creare un trigger per un argomento di Amazon SNS](#)
- [Guida per sviluppatori di Amazon Simple Notification Service](#)

Esempi di integrazione dalla community

Le seguenti sezioni forniscono collegamenti a post di blog, articoli ed esempi della community.

Note

Questi collegamenti sono forniti solo a scopo informativo e non devono essere considerati né un elenco completo né un'approvazione del contenuto degli esempi. AWS non è responsabile per il contenuto o l'accuratezza dei contenuti esterni.

Argomenti

- [Post del blog](#)
- [Esempi di codice](#)

Post del blog

- [Integrazione SonarQube come Pull Request Approver su AWS CodeCommit](#)

Scopri come creare un CodeCommit repository che richieda un'analisi di SonarQube qualità corretta prima di poter unire le pull request.

Pubblicazione: 12 dicembre 2019

- [Migrazione verso AWS CodeCommit e AWS CodePipeline da AWS CodeBuild GitLab](#)

Scopri come migrare più repository AWS CodeCommit da GitLab e configurare una pipeline CI/CD utilizzando and. AWS CodePipeline AWS CodeBuild

Pubblicazione: 22 novembre 2019

- [Implementazione utilizzando GitFlow , e AWS CodePipelineAWS CodeCommitAWS CodeBuildAWS CodeDeploy](#)

Scopri come implementare GitFlow utilizzando AWS CodePipeline, AWS CodeCommit AWS CodeBuild, e AWS CodeDeploy.

Pubblicazione: 22 febbraio 2019

- [Usare Git AWS CodeCommit con più AWS account](#)

Scopri come gestire la configurazione Git su più account Amazon Web Services.

Pubblicazione: 12 febbraio 2019

- [Convalida delle richieste AWS CodeCommit pull con e AWS CodeBuildAWS Lambda](#)

Scopri come convalidare le richieste pull con AWS CodeCommit AWS CodeBuild, e. AWS Lambda Eseguendo dei test sulle modifiche proposte prima di unirle nel ramo predefinito, puoi contribuire a garantire un elevato livello di qualità nelle pull request, a catturare eventuali problemi potenziali e ad aumentare la fiducia dello sviluppatore in relazione alle modifiche apportate.

Pubblicazione: 11 febbraio 2019

- [Utilizzo di identità federate con AWS CodeCommit](#)

Scopri come accedere agli archivi AWS CodeCommit utilizzando le identità utilizzate nella tua azienda.

Pubblicazione: 5 ottobre 2018

- [Perfezionamento dell'accesso alle filiali in AWS CodeCommit](#)

Scopri come limitare i commit ai branch del repository creando e applicando una policy IAM che utilizza una chiave di contesto.

Pubblicazione: 16 maggio 2018

- [Replica dei AWS CodeCommit repository tra regioni utilizzando Fargate AWS](#)

Scopri come configurare la replica continua di un CodeCommit repository da una AWS regione all'altra utilizzando un'architettura serverless.

Pubblicazione: 11 aprile 2018

- [Distribuzione dell'infrastruttura AWS OpsWorks for Chef Automate](#)

Scopri come utilizzare CodePipeline e come garantire che CodeBuild i libri AWS Lambda di cucina e altre configurazioni vengano distribuiti in modo coerente su due o più server Chef residenti in uno o più. CodeCommit Regioni AWS

Pubblicazione: 9 marzo 2018

- [Burro di arachidi e cioccolato: pipeline CI/CD delle funzioni Azure con AWS CodeCommit](#)

Scopri come creare una pipeline CI/CD PowerShell basata su Azure Functions in cui il codice è archiviato in un repository. CodeCommit

Pubblicazione: 19 febbraio 2018

- [Distribuzione continua su Kubernetes utilizzando AWS CodePipeline, AWS CodeCommit, AWS CodeBuild Amazon ECR e AWS Lambda](#)

Scopri come usare Kubernetes e come creare AWS insieme una pipeline di distribuzione continua e completamente gestita per applicazioni basate su container.

Pubblicazione: 11 gennaio 2018

- [Usa le AWS CodeCommit Pull Requests per richiedere revisioni del codice e discutere del codice](#)

Scopri come utilizzare le richieste pull per rivedere, commentare e iterare in modo interattivo sulle modifiche al codice in un CodeCommit repository.

Pubblicazione: 20 novembre 2017

- [Crea AWS CodeCommit flussi di lavoro serverless utilizzando Amazon CloudWatch Events e JGit](#)

Scopri come creare regole relative CloudWatch agli eventi che elaborano le modifiche in un repository utilizzando gli eventi del CodeCommit repository e indirizzano le azioni in altri servizi. AWS Gli esempi includono AWS Lambda funzioni che applicano le policy dei messaggi di commit Git sui commit, replicano un CodeCommit repository ed eseguono il backup di un repository su Amazon S3. CodeCommit

Pubblicazione: 3 agosto 2017

- [Migrazione verso AWS CodeCommit](#)

Scopri come inviare codice a due repository come parte della migrazione dall'uso di un altro repository Git a quando lo usi. CodeCommit SourceTree

Pubblicazione: 6 settembre 2016

- [Imposta test continui con Appium, Jenkins e AWS CodeCommitAWS Device Farm](#)

Scopri come creare un processo di test continuo per i dispositivi mobili utilizzando Appium CodeCommit, Jenkins e Device Farm.

Pubblicazione: 2 febbraio 2016

- [Utilizzo AWS CodeCommit con repository Git in più account Amazon Web Services](#)

Scopri come clonare il tuo CodeCommit repository e, con un solo comando, configura l'helper delle credenziali per utilizzare un ruolo IAM specifico per le connessioni a quel repository.

Pubblicazione: novembre 2015

- [Integrazione e OpsWorksAWS CodeCommit](#)

Scopri come OpsWorks recuperare automaticamente app e libri di cucina Chef da. CodeCommit

Pubblicazione: 25 agosto 2015

- [Utilizzo AWS CodeCommit e credenziali di GitHub aiuto](#)

Scopri come configurare il tuo file gitconfig in modo che funzioni sia con gli helper per le credenziali. CodeCommit GitHub

Pubblicazione: settembre 2015

- [AWS CodeCommit Utilizzo da Eclipse](#)

Scopri come utilizzare gli EGit strumenti di Eclipse con cui lavorare. CodeCommit

Pubblicazione: agosto 2015

- [AWS CodeCommit con Amazon EC2 Role Credentials](#)

Scopri come utilizzare un profilo di istanza per Amazon EC2 durante la configurazione dell'accesso automatizzato degli agenti a un CodeCommit repository.

Pubblicazione: luglio 2015

- [Integrazione con Jenkins AWS CodeCommit](#)

Scopri come utilizzare CodeCommit e Jenkins per supportare due semplici scenari di integrazione continua (CI).

Pubblicazione: luglio 2015

- [Integrazione AWS CodeCommit con Review Board](#)

Scopri come integrarti CodeCommit in un flusso di lavoro di sviluppo utilizzando il sistema di [revisione del codice di Review Board](#).

Pubblicazione: luglio 2015

Esempi di codice

Di seguito sono riportati esempi di codice che potrebbero interessare CodeCommit gli utenti.

- [Script Mac OS X per l'eliminazione periodica delle credenziali memorizzate nella cache dello store certificati OS X](#)

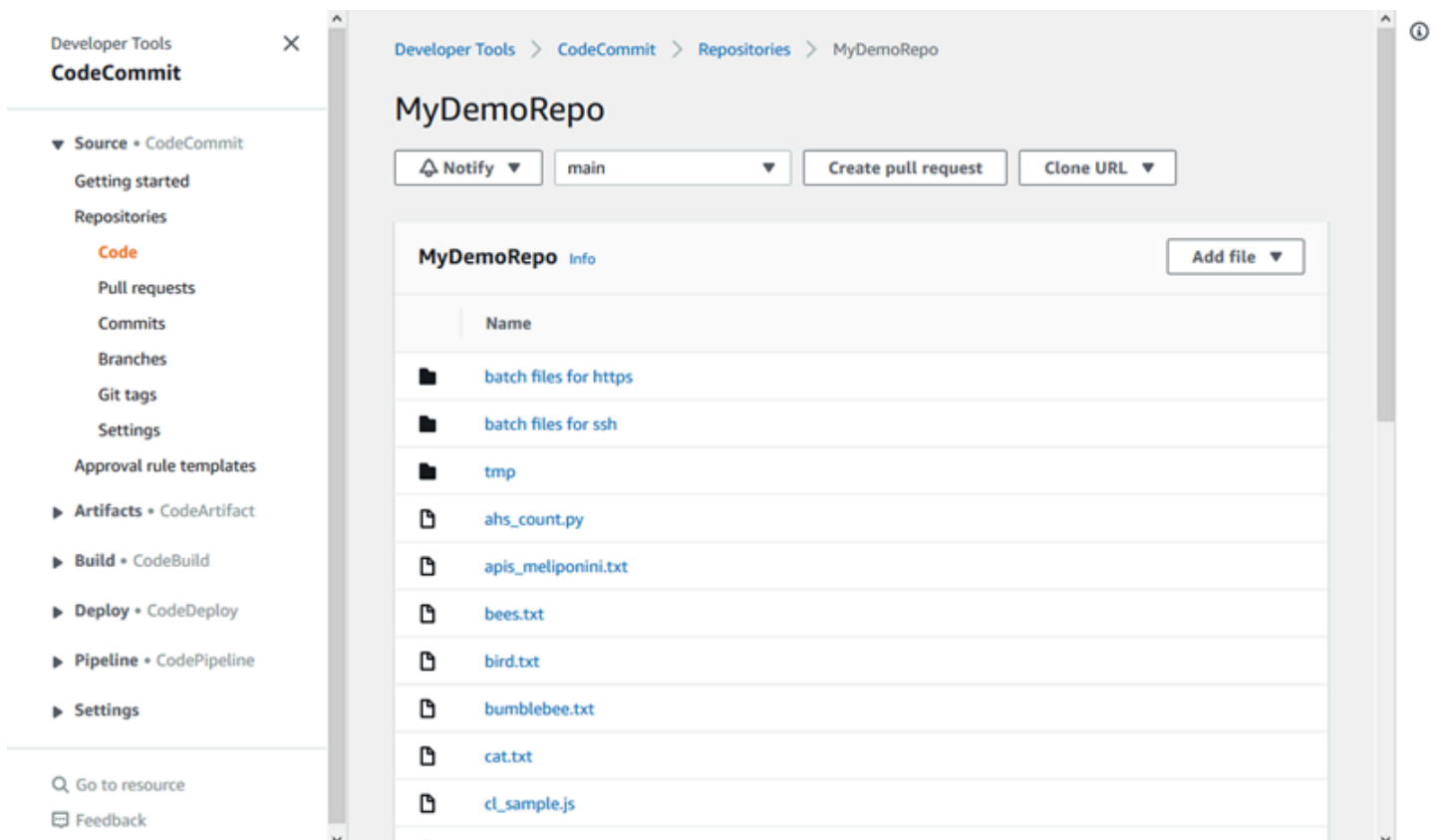
Se utilizzate l'helper per le credenziali per CodeCommit Mac OS X, probabilmente conoscete il problema delle credenziali memorizzate nella cache. Questo script mostra una soluzione.

Autore: Nico Coetzee

Pubblicazione: febbraio 2016

Lavorare con i repository in AWS CodeCommit

Un repository è l'oggetto di controllo della versione fondamentale in CodeCommit. Qui puoi archiviare in modo sicuro il codice e i file del progetto. Puoi anche archiviare la cronologia del progetto, dal primo commit fino alle ultime modifiche. Puoi condividere il repository con altri utenti in modo da collaborare a un progetto. Se si aggiungono AWS tag ai repository, è possibile impostare notifiche in modo che gli utenti del repository ricevano e-mail sugli eventi (ad esempio, un altro utente che commenta il codice). Puoi anche modificare le impostazioni di default del repository, sfogliarne i contenuti e molto altro. Puoi creare trigger per il repository in modo che i push del codice o altri eventi attivino azioni, come e-mail o funzioni del codice. Puoi persino configurare un repository sul computer locale (repository locale) per eseguire il push delle modifiche in più repository.



Prima di poter inviare modifiche a un CodeCommit repository, devi configurare un utente IAM nel tuo account Amazon Web Services o configurare l'accesso per l'accesso federato o le credenziali temporanee. Per ulteriori informazioni, consultare [Fase 1: Configurazione iniziale per CodeCommit](#) e [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Per informazioni sull'utilizzo di altri aspetti del repository in CodeCommit, consulta,, [Lavorare con i file](#)[Utilizzo delle richieste pull](#), [Lavorare con i commit](#) e. [Gestione dei rami](#) [Utilizzo delle preferenze dell'utente](#) Per informazioni sulla migrazione a CodeCommit, consulta. [Migrare a CodeCommit](#)

Argomenti

- [Creare un repository AWS CodeCommit](#)
- [Connect a un AWS CodeCommit repository](#)
- [Condividi un repository AWS CodeCommit](#)
- [Configurazione delle notifiche per gli eventi in un repository AWS CodeCommit](#)
- [Taggare i repository in AWS CodeCommit](#)
- [Gestisci i trigger per un repository AWS CodeCommit](#)
- [Associare o dissociare un AWS CodeCommit repository con Amazon Reviewer CodeGuru](#)
- [Visualizza i dettagli del CodeCommit repository](#)
- [Modifica le impostazioni del AWS CodeCommit repository](#)
- [Sincronizza le modifiche tra un repository locale e un repository AWS CodeCommit](#)
- [Invia i commit a un repository Git aggiuntivo](#)
- [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#)
- [Elimina un repository AWS CodeCommit](#)

Creare un repository AWS CodeCommit

Usa la AWS CodeCommit console o il AWS Command Line Interface (AWS CLI) per creare un CodeCommit repository vuoto. Per aggiungere i tag a un repository dopo la sua creazione, consulta [Aggiunta di un tag a un repository](#).

Queste istruzioni presuppongono che la procedura in [Configurazione](#) sia stata completata.

Note

A seconda dell'utilizzo, potrebbero esserti addebitati costi per la creazione o l'accesso a un repository. Per ulteriori informazioni, consulta la sezione [Prezzi](#) nella pagina di informazioni sul CodeCommit prodotto.

Argomenti

- [Crea un repository \(console\)](#)
- [Crea un repository \(AWS CLI\)](#)

Crea un repository (console)

Per creare un repository CodeCommit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove vuoi creare il repository. Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository), scegli Create repository (Crea repository).
4. Nella pagina Create repository (Crea repository) immettere un nome per il repository in Repository name (Nome repository).

Note

I nomi dei repository distinguono tra maiuscole e minuscole. Il nome deve essere univoco nel Regione AWS tuo account Amazon Web Services.

5. (Facoltativo) In Description (Descrizione), immettere una descrizione per il repository. Ciò può consentire agli utenti di identificare lo scopo del repository.

Note

Il campo della descrizione mostra Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza GetRepository o BatchGetRepositories APIs e prevedi di visualizzare il campo di descrizione del repository in un browser web, consulta l'CodeCommit API Reference.](#)

6. (Facoltativo) Scegliete Aggiungi tag per aggiungere uno o più tag del repository (un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse) al repository. Per ulteriori informazioni, consulta [Taggare i repository in AWS CodeCommit](#).
7. (Facoltativo) Espandi la configurazione aggiuntiva per specificare se utilizzare la chiave predefinita Chiave gestita da AWS o la chiave gestita dal cliente per crittografare e decrittografare i dati in questo repository. Se scegli di utilizzare la tua chiave gestita dal cliente, devi assicurarti che sia disponibile nel luogo in Regione AWS cui stai creando il repository e

che la chiave sia attiva. Per ulteriori informazioni, consulta [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#).

8. (Facoltativo) Seleziona Abilita Amazon CodeGuru Reviewer per Java e Python se questo repository contiene codice Java o Python e desideri che CodeGuru Reviewer lo analizzi. CodeGuru Reviewer utilizza diversi modelli di machine learning per trovare difetti nel codice e suggerire miglioramenti e correzioni nelle pull request. Per ulteriori informazioni, consulta la [Amazon CodeGuru Reviewer User Guide](#).
9. Scegli Create (Crea).

Dopo aver creato un repository, puoi connetterti ad esso e iniziare ad aggiungere codice tramite la CodeCommit console o un client Git locale, oppure integrando il tuo CodeCommit repository con il tuo IDE preferito. Per ulteriori informazioni, consulta [Configurazione per AWS CodeCommit](#). È anche possibile aggiungere il repository a una pipeline di distribuzione continua. Per ulteriori informazioni, consulta la sezione relativa alla [procedura guidata per una pipeline semplice](#).

Per ottenere informazioni sul nuovo CodeCommit repository, ad esempio quelle URLs da utilizzare per la clonazione del repository, scegli il nome del repository dall'elenco o scegli semplicemente il protocollo di connessione che desideri utilizzare accanto al nome del repository.

Per condividere questo repository con altri utenti, devi inviare loro il link HTTPS o SSH da utilizzare per clonare il repository. Verifica che dispongano delle autorizzazioni necessarie per accedere al repository. Per ulteriori informazioni, consultare [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Crea un repository (AWS CLI)

È possibile utilizzare il AWS CLI per creare un CodeCommit repository. A differenza della console, è possibile aggiungere i tag a un repository se viene creato con l' AWS CLI.

1. Assicurati di averlo configurato AWS CLI con il Regione AWS luogo in cui esiste il repository. Per verificare la regione, eseguire il comando seguente nella riga di comando o nel terminale e rivedere le informazioni sul nome di regione predefinito.

```
aws configure
```

Il nome della regione predefinito deve corrispondere a quello Regione AWS del repository in. CodeCommit Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).

2. Eseguire il comando `create-repository` specificando:

- Un nome che identifica in modo univoco il CodeCommit repository (con l'opzione). `--repository-name`

Note

Questo nome deve essere univoco per un account Amazon Web Services.

- Un commento facoltativo sul CodeCommit repository (con l'opzione `--repository-description`).
- Una o più coppie chiave-valore opzionali da usare come tag per il CodeCommit repository (con l'opzione). `--tags`
- Una chiave opzionale gestita dal cliente da utilizzare per la crittografia e la decrittografia di questo repository. Tutti gli archivi sono crittografati in transito e a riposo utilizzando una chiave in AWS KMS. Se non viene specificata alcuna chiave, `aws/codecommit` viene utilizzata la chiave AWS gestita predefinita.

Ad esempio, per creare un CodeCommit repository denominato `MyDemoRepo` con la descrizione `"My demonstration repository"` e un tag con una chiave denominata `Team` con il valore di `Saanvi`, usa questo comando.

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

Note

Il campo della descrizione visualizza Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza `GetRepository` o `BatchGetRepositories` APIs e prevedi di visualizzare il campo di descrizione del repository in un browser web, consulta l'`CodeCommit API Reference`.](#)

3. In caso di esito positivo, l'output del comando è un oggetto `repositoryMetadata` con le informazioni riportate di seguito:
 - Descrizione (`repositoryDescription`).
 - ID univoco generato dal sistema (`repositoryId`).

- Nome (repositoryName).
- L'ID dell'account Amazon Web Services associato al CodeCommit repository (accountId).

Di seguito è riportato un esempio di output in base al comando di esempio precedente:

```
{
  "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

I tag che sono stati aggiunti quando il repository è stato creato non vengono restituiti nell'output. Per visualizzare una lista di tag associati a un repository, eseguire il comando [list-tags-for-resource](#).

4. Prendi nota del nome e dell'ID del CodeCommit repository. Ne hai bisogno per monitorare e modificare le informazioni sul CodeCommit repository, specialmente se lo usi. AWS CLI

Se si dimentica il nome o l'ID, attenersi alle istruzioni in [Visualizza i dettagli CodeCommit del repository \(\)AWS CLI](#).

Dopo aver creato un repository, è possibile connettersi a esso e iniziare ad aggiungere codice. Per ulteriori informazioni, consulta [Connessione a un repository](#). È anche possibile aggiungere il repository a una pipeline di distribuzione continua. Per ulteriori informazioni, consulta la sezione relativa alla [procedura guidata per una pipeline semplice](#).

Connect a un AWS CodeCommit repository

Quando ci si connette a un CodeCommit repository per la prima volta, in genere si clona il contenuto sul computer locale. È inoltre possibile [aggiungere e modificare file](#) in un repository direttamente dalla console. CodeCommit In alternativa, se hai già un repository locale, puoi aggiungere un repository CodeCommit come remoto. Questo argomento fornisce istruzioni per la connessione a un CodeCommit repository. Se desideri migrare un repository esistente in, consulta. CodeCommit [Migrare a CodeCommit](#)

Note

A seconda dell'utilizzo, potrebbero esserti addebitati costi per la creazione o l'accesso a un repository. Per ulteriori informazioni, consulta la sezione [Prezzi](#) nella pagina di informazioni sul CodeCommit prodotto.

Argomenti

- [Prerequisiti per la connessione a un repository CodeCommit](#)
- [Connect al CodeCommit repository clonando il repository](#)
- [Connect un repository locale al CodeCommit repository](#)

Prerequisiti per la connessione a un repository CodeCommit

Prima di poter clonare un CodeCommit repository o connettere un repository locale a un repository: CodeCommit

- È necessario aver configurato il tuo computer locale con il software e le impostazioni necessarie per connettersi ad CodeCommit. Ciò include l'installazione e la configurazione di Git. Per ulteriori informazioni, consultare [Configurazione](#) e [Guida introduttiva a Git e AWS CodeCommit](#).
- È necessario disporre dell'URL clone del CodeCommit repository a cui si desidera connettersi. Per ulteriori informazioni, consulta [Visualizza i dettagli del repository](#).

Se non hai ancora creato un CodeCommit repository, segui le istruzioni riportate in [Creazione di un repository](#), copia l'URL del clone del CodeCommit repository e torna a questa pagina.

Se disponi di un CodeCommit repository ma non ne conosci il nome, segui le istruzioni riportate in. [Visualizza i dettagli del repository](#)

- È necessario disporre di una posizione sul computer locale in cui archiviare una copia locale del CodeCommit repository a cui ci si connette. (Questa copia locale del CodeCommit repository è nota come repository locale.) A questo punto passi alla copia locale ed esegui i comandi Git da tale posizione. Ad esempio, è possibile utilizzare `/tmp` (per Linux, macOS o Unix) o `c:\temp` (per Windows) se si sta creando un clone temporaneo a scopo di test. Questo è il percorso di directory utilizzato in questi esempi.

Note

È possibile usare qualsiasi directory. Se stai clonando un repository per l'uso a lungo termine, crea il clone da una directory di lavoro e non da una utilizzata per i file temporanei. Se utilizzi una directory differente da `/tmp` oppure da `c:\temp`, assicurati di sostituire la directory con la nostra quando esegui queste istruzioni.

Connect al CodeCommit repository clonando il repository

Se non disponi già di un repository locale, segui i passaggi di questa procedura per clonare il CodeCommit repository sul tuo computer locale.

1. Completare i prerequisiti, tra cui [Configurazione](#).

Important

Se non è stata completata la configurazione, non è possibile connettersi al repository o clonarlo.

2. Nella directory `/tmp` o `c:\temp`, utilizzare Git per eseguire il comando clone. Gli esempi seguenti mostrano come clonare un repository denominato *MyDemoRepo* nella regione Stati Uniti orientali (Ohio).

Per HTTPS con [credenziali Git](#) o l'assistente credenziali incluso con l' AWS CLI:

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

Per l'utilizzo di HTTPS[git-remote-codecommit](#), presupponendo il profilo predefinito e Regione AWS configurato in: AWS CLI

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

Per SSH:

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

In questo esempio, `git-codecommit.us-east-2.amazonaws.com` è il punto di connessione Git per la regione degli Stati Uniti orientali (Ohio) in cui esiste il repository, `MyDemoRepo` rappresenta il nome del CodeCommit repository e `my-demo-repo` rappresenta il nome della directory che Git crea nella `/tmp` directory o nella directory. `c:\temp` Per ulteriori informazioni sul Regioni AWS supporto CodeCommit e sulle relative connessioni Git Regioni AWS, vedere [Regioni ed endpoint di connessione Git](#).

Note

Quando si utilizza SSH su sistemi operativi Windows per clonare un repository, è necessario aggiungere l'ID chiave SSH alla stringa di connessione come segue:

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

Per ulteriori informazioni, consultare [Per le connessioni SSH su Windows](#) e [Risoluzione dei problemi](#).

Dopo che Git ha creato la directory, scarica una copia del tuo CodeCommit repository nella directory appena creata.

Se il CodeCommit repository è nuovo o comunque vuoto, viene visualizzato un messaggio che indica che stai clonando un repository vuoto. Si tratta di un comportamento normale.

Note

Se ricevi un errore che indica che Git non riesce a trovare il CodeCommit repository o che non disponi dell'autorizzazione per connetterti al CodeCommit repository, assicurati di aver completato i [prerequisiti](#), tra cui l'assegnazione delle autorizzazioni all'utente

IAM e la configurazione delle credenziali utente IAM per Git e sul computer locale. CodeCommit Inoltre, occorre accertarsi che il nome del repository specificato sia corretto.

Dopo aver collegato con successo il repository locale al CodeCommit repository, ora sei pronto per iniziare a eseguire i comandi Git dal repository locale per creare commit, branch e tag da inviare e scaricare dal repository. CodeCommit

Connect un repository locale al CodeCommit repository

Completa la procedura seguente se disponi già di un repository locale e vuoi aggiungere un repository CodeCommit come repository remoto. Se disponi già di un repository remoto e desideri inviare i tuoi commit a CodeCommit un altro repository remoto, segui la procedura riportata di seguito.

[Invia i commit a due repository](#)

1. Completare i [prerequisiti](#).
2. Dal prompt dei comandi o dal terminale, passa alla directory del repository locale ed esegui il git remote add comando per aggiungere il repository come repository remoto per il CodeCommit repository locale.

Ad esempio, il comando seguente aggiunge il telecomando soprannominato a mit.us-east-2.amazonaws. **origin** https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo:

Per HTTPS:

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

Per SSH:

```
git remote add origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Questo comando non restituisce alcun risultato.

3. Per verificare di aver aggiunto il CodeCommit repository come telecomando per il repository locale, esegui il git remote -v comando, che dovrebbe creare un output simile al seguente:

Per 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)
```

Per 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)
```

Dopo aver collegato con successo il repository locale al CodeCommit repository, sei pronto per iniziare a eseguire i comandi Git dal repository locale per creare commit, branch e tag e per eseguire il push e il pull dal repository. CodeCommit

Condividi un repository AWS CodeCommit

Dopo aver creato un CodeCommit repository, puoi condividerlo con altri utenti. Innanzitutto, stabilisci se utilizzerai un accesso federato, credenziali temporanee o un provider di identità web come IAM Identity Center per l'accesso CodeCommit, o se desideri utilizzare credenziali Git o coppie di chiavi SSH con utenti IAM. Se utilizzi il primo, dovrai configurare gli utenti, l'accesso e le autorizzazioni per il tuo provider di identità e quindi fornire istruzioni per l'uso da parte degli utenti. git-remote-codecommit Per ulteriori informazioni, consultare [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#) e [Connessione ai AWS CodeCommit repository con credenziali rotanti](#).

Non è possibile utilizzare credenziali Git o coppie di chiavi SSH con provider di accesso federato o di identità, ma molti IDEs funzionano meglio con queste credenziali. In questo caso, decidi quale protocollo (HTTPS o SSH) consigliare agli utenti durante la clonazione e l'utilizzo di un client Git o un IDE per connettersi al tuo repository. Quindi inviare l'URL e le informazioni di connessione agli utenti con i quali si desidera condividere il repository. A seconda dei requisiti di sicurezza, la condivisione di un repository potrebbe richiedere anche la creazione di un gruppo IAM, l'applicazione di policy gestite a quel gruppo e la modifica delle policy IAM per perfezionare l'accesso o la creazione e l'utilizzo di ruoli IAM.

 Note

Dopo aver concesso agli utenti l'accesso console al repository, questi possono aggiungere o modificare i file direttamente nella console senza dover configurare un client Git o un'altra connessione. Per ulteriori informazioni, consultare [Creare o aggiungere un file a un AWS CodeCommit repository](#) e [Modifica il contenuto di un file in un AWS CodeCommit repository](#).

Queste istruzioni presuppongono che sia già stata completata la procedura in [Configurazione](#) e [Creazione di un repository](#).

 Note

A seconda dell'utilizzo, potrebbero esserti addebitati costi per la creazione o l'accesso a un repository. Per ulteriori informazioni, consulta la sezione [Prezzi](#) nella pagina di informazioni sul CodeCommit prodotto.

Argomenti

- [Scegli il protocollo di connessione da condividere con i tuoi utenti](#)
- [Crea policy IAM per il tuo repository](#)
- [Crea un gruppo IAM per gli utenti del repository](#)
- [Condividi le informazioni di connessione con i tuoi utenti](#)

Scegli il protocollo di connessione da condividere con i tuoi utenti

Quando crei un repository in CodeCommit, vengono generati due endpoint: uno per le connessioni HTTPS e uno per le connessioni SSH. Entrambi offrono connessioni protette in rete. Gli utenti possono scegliere quale dei due protocolli utilizzare. Entrambi gli endpoint rimangono attivi indipendentemente dal protocollo consigliato per gli utenti.

Le connessioni HTTPS richiedono:

- Credenziali Git, che gli utenti IAM possono generare autonomamente in IAM. Le credenziali Git sono il metodo più semplice di configurazione e utilizzo per gli utenti del repository.
- Una chiave o un ruolo di AWS accesso da assumere, che gli utenti del repository devono configurare nel proprio profilo di credenziali. È possibile configurare git-remote-codecommit (scelta

consigliata) o l'assistente credenziali incluso nell' AWS CLI. Questi sono gli unici metodi disponibili per l'account root o gli utenti federati.

Per le connessioni SSH gli utenti devono:

- Generare una coppia di chiavi pubblica-privata.
- Archiviare la chiave pubblica.
- Associa la chiave pubblica al loro utente IAM.
- Configurare i relativi file host noti sul computer locale.
- Creare e gestire un file di configurazione sui computer locali.

Poiché si tratta di un processo di configurazione più complesso, ti consigliamo di scegliere credenziali HTTPS e Git per le connessioni a CodeCommit.

Per ulteriori informazioni su HTTPS, SSH, Git, git-remote-codecommit e repository remoti, consulta [Configurazione](#), [Connessione ai AWS CodeCommit repository con credenziali rotanti](#) o la documentazione di Git.

Note

Anche se Git supporta un'ampia gamma di protocolli di connessione, CodeCommit non supporta le connessioni con protocolli non protetti, ad esempio il protocollo locale o HTTP generico.

Crea policy IAM per il tuo repository

AWS fornisce tre policy gestite in IAM per CodeCommit. Queste politiche non possono essere modificate e si applicano a tutti gli archivi associati al tuo account Amazon Web Services. Tuttavia, è possibile utilizzare questi criteri come modelli per creare policy gestite personalizzate che si applicano solo al repository da condividere. Le policy gestite dei tuoi clienti possono applicarsi nello specifico al repository che desideri condividere. Per ulteriori informazioni, consulta [Managed Policies](#) e [IAM Users and Groups](#).

 Tip

Per un controllo più preciso sull'accesso al tuo repository, puoi creare più di una policy gestita dal cliente e applicare le policy a diversi utenti e gruppi IAM.

Per informazioni sulla revisione dei contenuti di policy gestite e l'utilizzo di policy per creare e applicare autorizzazioni, consulta [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Creazione di una policy gestita dal cliente per il repository

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nell'area di navigazione Dashboard (Pannello di controllo) scegliere Policies (Policy) e quindi Crea Policy (Crea policy).
3. Nella pagina Crea policy, scegli Importa policy gestita.
4. Nella pagina Importa policy gestite, in Filtra policy, inserisci **AWSCodeCommitPowerUser**. Scegli il pulsante accanto al nome della politica, quindi scegli Importa.
5. Nella pagina Create policy (Crea policy), scegli JSON. Sostituisci la parte «*» della Resource riga relativa CodeCommit alle azioni con l'Amazon Resource Name (ARN) del CodeCommit repository, come illustrato di seguito:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

 Tip

Per trovare l'ARN per il CodeCommit repository, vai alla CodeCommit console, scegli il nome del repository dall'elenco, quindi scegli Impostazioni. Per ulteriori informazioni, consulta [Visualizza i dettagli del repository](#).

Per fare in modo che questa policy venga applicata a più repository, aggiungere ogni repository come una risorsa, specificandone l'ARN. Includere una virgola tra ciascuna dichiarazione di risorsa, come nell'esempio seguente:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Quando hai finito di modificare, scegli Review policy.

6. Nella pagina Rivedi politica, in Nome, inserisci un nuovo nome per la politica (ad esempio, *AWSCodeCommitPowerUser-MyDemoRepo*). Facoltativamente, fornisci una descrizione per questa politica.
7. Scegli Crea policy.

Crea un gruppo IAM per gli utenti del repository

Per gestire l'accesso al tuo repository, crea un gruppo IAM per i relativi utenti, aggiungi gli utenti IAM a quel gruppo e quindi allega la policy gestita dai clienti che hai creato nel passaggio precedente. In alternativa, puoi creare un ruolo con una policy gestita dal cliente allegata e far sì che gli utenti assumano quel ruolo.

Se utilizzi SSH, devi allegare un'altra policy gestita al IAMUser SSHKeys gruppo, la policy gestita da IAM che consente agli utenti di caricare la propria chiave pubblica SSH e associarla all'utente IAM a cui utilizzano per connettersi. CodeCommit

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nell'area di navigazione Dashboard (Pannello di controllo), scegliere Groups (Gruppi) e quindi Create New Group (Crea nuovo gruppo).
3. Nella pagina Imposta nome gruppo, in Nome gruppo, inserisci un nome per il gruppo (ad esempio, *MyDemoRepoGroup*), quindi scegli Passaggio successivo. Valutare la possibilità di includere il nome del repository nel nome del gruppo.

Note

Questo nome deve essere univoco per un account Amazon Web Services.

4. Seleziona la casella accanto alla politica gestita dai clienti che hai creato nella sezione precedente (ad esempio, *AWSCodeCommitPowerUser-MyDemoRepo*).

5. Nella pagina Review (Verifica), scegliere Create Group (Crea gruppo). IAM crea questo gruppo con le politiche specificate già allegate. Il gruppo viene visualizzato nell'elenco dei gruppi associati al tuo account Amazon Web Services.
6. Selezionare il gruppo dall'elenco.
7. Nella pagina di riepilogo del gruppo, scegliere la scheda Users (Utenti) e quindi scegliere Add Users to Group (Aggiungi utenti al gruppo). Nell'elenco che mostra tutti gli utenti associati al tuo account Amazon Web Services, seleziona le caselle accanto agli utenti a cui desideri consentire l'accesso al CodeCommit repository, quindi scegli Aggiungi utenti.

 Tip

È possibile utilizzare la casella di ricerca per trovare rapidamente gli utenti per nome.

8. Dopo aver aggiunto gli utenti, chiudi la console IAM.

Condividi le informazioni di connessione con i tuoi utenti

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository) scegliere il repository che si desidera eliminare.
4. In Clone URL (URL clone), scegliere il protocollo che deve essere utilizzato dagli utenti. L'URL clone viene copiato per il protocollo di connessione scelto.
5. Invia ai tuoi utenti l'URL del clone insieme a qualsiasi altra istruzione, come l'installazione AWS CLI, la configurazione di un profilo o l'installazione di Git. Assicurarsi di includere le informazioni di configurazione per il protocollo di connessione (ad esempio HTTPS).

L'e-mail di esempio seguente fornisce informazioni per gli utenti che si connettono al MyDemoRepo repository con il protocollo di connessione HTTPS e le credenziali Git nella regione Stati Uniti orientali (Ohio) (us-east-2). Questa e-mail presuppone che l'utente abbia già installato Git e sappia utilizzarlo.

```
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](#).

È possibile trovare le istruzioni complete per la configurazione in. [Configurazione](#)

Configurazione delle notifiche per gli eventi in un repository AWS CodeCommit

Puoi configurare le regole delle notifiche per un repository in modo che gli utenti ricevano e-mail sui tipi di evento di repository specificati. Le notifiche vengono inviate quando gli eventi corrispondono alle impostazioni della regola della notifica. Puoi creare un argomento Amazon SNS da utilizzare per le notifiche o utilizzarne uno esistente nel tuo account Amazon Web Services. Puoi utilizzare la CodeCommit console e configurare AWS CLI le regole di notifica.

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

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 noneSelect all

Comments	Pull request	Branches and tags
☐ On Commits	☒ Source Updated	☐ Created
☒ On Pull requests	☒ Created	☒ Deleted
	☒ Status Changed	☐ Updated
	☒ Merged	

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

× Remove target

Add target

Create SNS topic

CancelSubmit

Argomenti

- [Utilizzo delle regole di notifica del repository](#)
- [Creazione di una regola di notifica](#)

- [Modificare o disabilitare le notifiche](#)
- [Eliminare le notifiche](#)

Utilizzo delle regole di notifica del repository

La configurazione delle regole di notifica aiuta gli utenti del repository inviando e-mail quando qualcuno richiede un'operazione che interessa un altro utente. Ad esempio, puoi configurare una regola di notifica per inviare notifiche quando i commenti vengono effettuati sui commit. In questa configurazione, quando un utente del repository commenta una riga di codice in un commit, gli altri utenti del repository riceveranno un'e-mail. Possono accedere e visualizzare il commento. Anche le risposte ai commenti generano e-mail, quindi gli utenti del repository restano informati.

Le regole di notifica sono diverse dai trigger del repository e sono anche diverse dalle notifiche che potevi configurare nella CodeCommit console prima del 5 novembre 2019.

- Sebbene sia possibile configurare un trigger per utilizzare Amazon SNS per inviare e-mail su alcuni eventi del repository, tali eventi sono limitati a eventi operativi, come la creazione di filiali e l'invio di codice a una filiale. I trigger non utilizzano le regole CloudWatch Events per valutare gli eventi del repository. Sono più limitati all'ambito. Per ulteriori informazioni sull'uso dei trigger, vedi [Gestione dei trigger per un repository](#).
- Le notifiche configurate prima del 5 novembre 2019 avevano meno tipi di eventi disponibili e non potevano essere configurate per l'integrazione con le chat room di Amazon Chime o i canali Slack. Puoi continuare a utilizzare le notifiche configurate prima del 5 novembre 2019, ma non puoi creare notifiche di questo tipo. Al contrario, crea e utilizza le regole di notifica. Ti consigliamo di utilizzare le regole di notifica e di disabilitare o eliminare le notifiche create prima del 5 novembre 2019. Per ulteriori informazioni, consultare [Creazione di una regola di notifica](#) e [Eliminare le notifiche](#).

Creazione di una regola di notifica

È possibile utilizzare le regole di notifica per notificare agli utenti modifiche importanti, ad esempio quando viene creata una richiesta pull in un repository. Le regole di notifica specificano sia gli eventi che l'argomento Amazon SNS utilizzato per inviare le notifiche. Per ulteriori informazioni, vedere [Cosa sono le notifiche?](#)

 Note

Questa funzionalità non è disponibile nella regione Europa (Milano). Per informazioni su come configurare le notifiche nell'esperienza disponibile in quella regione, consulta [Configurare le notifiche del repository](#).

Puoi utilizzare la console o AWS CLI creare regole di notifica per AWS CodeCommit.

Per creare una regola di notifica (console)

1. Accedi a Console di gestione AWS e apri la CodeCommit console all'indirizzo <https://console.aws.amazon.com/codecommit/>.
2. Scegliere Repositories (Repository), quindi scegliere un repository in cui aggiungere regole di notifica.
3. Nella pagina delle risorse, scegliere Notify (Notifica), quindi selezionare Create notification rule (Crea regola di notifica). È anche possibile andare alla pagina Settings (Impostazioni) del repository e scegliere Create notification rule (Crea regola di notifica).
4. In Notification name (Nome notifica), immettere un nome per la regola.
5. In Tipo di dettaglio, scegli Basic se desideri che nella notifica siano EventBridge incluse solo le informazioni fornite ad Amazon. Scegli Completo se desideri includere le informazioni fornite ad Amazon EventBridge e le informazioni che potrebbero essere fornite da CodeCommit o dal gestore delle notifiche.

Per ulteriori informazioni, vedere [Informazioni sul contenuto delle notifiche e sulla sicurezza](#).

6. In Events that trigger notifications (Eventi che attivano le notifiche), selezionare gli eventi per i quali si desidera inviare notifiche. Per ulteriori informazioni, consulta l'argomento relativo agli [eventi per le regole di notifica sui repository](#).
7. In Targets (Destinazioni), procedere in uno dei seguenti modi:
 - Se hai già configurato una risorsa da utilizzare con le notifiche, in Scegli il tipo di destinazione, scegli Amazon Q Developer nelle applicazioni di chat (Slack) o l'argomento SNS. In Scegli destinazione, scegli il nome del client (per un client Slack configurato in Amazon Q Developer nelle applicazioni di chat) o l'Amazon Resource Name (ARN) dell'argomento Amazon SNS (per gli argomenti Amazon SNS già configurati con la politica richiesta per le notifiche).

- Se non è stata configurata una risorsa da utilizzare con le notifiche, scegliere Create target (Crea destinazione), e quindi scegliere SNS topic (Argomento SNS). Fornire un nome per l'argomento dopo codestar-notifications-, quindi scegliere Create (Crea).

Note

- Se si crea l'argomento Amazon SNS come parte della creazione della regola di notifica, viene applicata la policy che consente alla funzionalità di notifica di pubblicare eventi nell'argomento. L'utilizzo di un argomento creato per le regole di notifica consente di iscrivere solo gli utenti che si desidera ricevano le notifiche relative a questa risorsa.
- Non puoi creare un client Amazon Q Developer in chat come parte della creazione di una regola di notifica. Se scegli Amazon Q Developer nelle applicazioni di chat (Slack), vedrai un pulsante che ti indirizza a configurare un client in Amazon Q Developer nelle applicazioni di chat. Scegliendo questa opzione si apre la console delle applicazioni di chat di Amazon Q Developer. Per ulteriori informazioni, consulta [Configurare le integrazioni tra Notifications e Amazon Q Developer nelle applicazioni di chat](#).
- Se desideri utilizzare un argomento Amazon SNS esistente come obiettivo, devi aggiungere la politica richiesta per AWS CodeStar le notifiche oltre a qualsiasi altra politica che potrebbe esistere per quell'argomento. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione degli argomenti Amazon SNS per le notifiche](#) e le [informazioni sulla sicurezza e sui contenuti delle notifiche](#).

8. Per completare la creazione della regola, scegliere Invia.
9. È necessario iscrivere gli utenti all'argomento Amazon SNS relativo alla regola prima che possano ricevere notifiche. Per ulteriori informazioni, consulta [Sottoscrivere gli utenti agli argomenti di Amazon SNS relativi agli obiettivi](#). Puoi anche configurare l'integrazione tra le notifiche e Amazon Q Developer nelle applicazioni di chat per inviare notifiche alle chat room di Amazon Chime. Per ulteriori informazioni, consulta [Configurare l'integrazione tra Notifications e Amazon Q Developer nelle applicazioni di chat](#).

Per creare una regola di notifica (AWS CLI)

1. Da un terminale o dal prompt dei comandi, eseguire il comando create-notification rule per generare lo skeleton JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

È possibile assegnare al file un nome qualsiasi. In questo esempio, il file è denominato *rule.json*.

2. Aprire il file JSON in un editor di testo normale e modificarlo per includere la risorsa, i tipi di evento e la destinazione desiderata per la regola. L'esempio seguente mostra una regola di notifica denominata **MyNotificationRule** per un repository denominato *MyDemoRepo* in un AWS account con l'ID. *123456789012* Le notifiche con il tipo di dettaglio completo vengono inviate a un argomento di Amazon SNS denominato *MyNotificationTopic* quando vengono creati rami e tag:

```
{  
  "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"  
}
```

Salvare il file.

3. Utilizzando il file appena modificato, dal terminale o dalla riga di comando, eseguire nuovamente il comando `create-notification-rule` per creare la regola di notifica:

```
aws codestar-notifications create-notification-rule --cli-input-json  
file://rule.json
```

4. In caso di esito positivo, il comando restituisce l'ARN della regola di notifica, simile al seguente:

```
{
```

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

Modificare o disabilitare le notifiche

Puoi utilizzare la AWS CodeCommit console per modificare la modalità di configurazione delle notifiche create prima del 5 novembre 2019, inclusi i tipi di eventi che inviano e-mail agli utenti e l'argomento Amazon SNS utilizzato per inviare e-mail relative al repository. Puoi inoltre utilizzare la console CodeCommit per gestire l'elenco di indirizzi e-mail e di endpoint abbonati all'argomento o per disabilitare le notifiche.

Per modificare le impostazioni di notifica

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera configurare le notifiche create prima del 5 novembre 2019.
3. Nel riquadro di navigazione, scegliere Settings (Impostazioni), quindi Notifications (Notifiche). Se viene visualizzato un banner che informa che sono presenti notifiche anziché regole di notifica, scegliere Manage existing notifications (Gestisci notifiche esistenti).
4. Scegli Modifica.
5. Apportare le modifiche, quindi fare clic su Save (Salva).

La disabilitazione notifiche è un modo semplice per impedire temporaneamente agli utenti di ricevere e-mail sugli eventi del repository.

Per eliminare definitivamente una notifica creata prima del 5 novembre 2019, segui la procedura descritta in [Eliminare le notifiche](#).

Per disabilitare le notifiche

1. [Apri la CodeCommit console su codecommit/home. https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera disabilitare le notifiche.
3. Nel riquadro di navigazione, scegliere Settings (Impostazioni), quindi Notifications (Notifiche). Scegliere Manage existing notifications (Gestisci notifiche esistenti).

4. Scegliere Edit (Modifica), e in Event status (Stato evento) utilizzare il cursore per disattivare Enable notifications (Abilita notifiche). Scegli Save (Salva).
5. Lo stato dell'evento cambia in Disabled (Disabilitato). Non vengono inviate e-mail sugli eventi. Quando disabiliti le notifiche, la regola CloudWatch Events per il repository viene disabilitata automaticamente. Non modificarne manualmente lo stato nella console CloudWatch Events.

Eliminare le notifiche

Se non desideri più utilizzare le notifiche create per un repository prima del 5 novembre 2019, puoi eliminare la regola Amazon CloudWatch Events associata alla notifica. Questa operazione eliminerà automaticamente la notifica. Non elimina alcun abbonamento o l'argomento Amazon SNS utilizzato per le notifiche.

Note

Se modifichi il nome di un repository dalla console, le notifiche create prima del 5 novembre 2019 continuano a funzionare senza alcuna modifica. Tuttavia, se modifichi il nome del repository dalla riga di comando o utilizzando l'API, le notifiche non funzionano più. Il modo più semplice per ripristinare le notifiche è eliminare le impostazioni di notifica, quindi configurarle di nuovo.

Per eliminare le impostazioni di notifica

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories (Repository), scegliere il nome del repository in cui si desidera rimuovere le notifiche create prima del 5 novembre 2019.
3. Nel riquadro di navigazione, scegliere Settings (Impostazioni), quindi Notifications (Notifiche). Se viene visualizzato un banner che informa che sono presenti notifiche anziché regole di notifica, scegliere Manage existing notifications (Gestisci notifiche esistenti).
4. In CloudWatch caso di regola, copia il nome della regola creata per la notifica.
5. Accedi a Console di gestione AWS e apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
6. In Events (Eventi), scegliere Rules (Regole). In Name (Nome), incollare il nome della regola creata per la notifica. Scegliere la regola e in Actions (Operazioni), scegliere Delete (Elimina).

7. [\(Facoltativo\) Per modificare o eliminare l'argomento Amazon SNS utilizzato per le notifiche dopo aver eliminato le impostazioni di notifica, accedi alla console Amazon SNS in v3/home. <https://console.aws.amazon.com/sns/>](#) Per ulteriori informazioni, consulta [Clean Up](#) in [Amazon Simple Notification Service Developer Guide](#).

Taggare i repository in AWS CodeCommit

Un tag è un'etichetta di attributo personalizzata che l'utente o AWS assegna a una AWS risorsa. AWS i tag sono diversi dai tag Git, che possono essere applicati ai commit. Ogni AWS tag è composto da due parti:

- Una chiave di tag (ad esempio, CostCenter, Environment, Project o Secret). Le chiavi dei tag distinguono tra maiuscole e minuscole
- Un campo facoltativo noto come valore del tag (ad esempio, 111122223333, Production o un nome di team). Non specificare il valore del tag equivale a utilizzare una stringa vuota. Come le chiavi tag, i valori dei tag fanno distinzione tra maiuscole e minuscole.

Tutti questi sono noti come coppie chiave-valore. Per conoscere i possibili limiti sul numero di tag in un repository e le limitazioni di chiavi e valori di tag, consulta [Limiti](#).

I tag ti aiutano a identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a un CodeCommit repository che assegni a un bucket Amazon S3. [Per ulteriori informazioni sulle strategie di tagging, consulta Tagging Resources. AWS](#)

Nel CodeCommit, la risorsa principale è un repository. È possibile utilizzare la CodeCommit console, o AWS SDKs per aggiungere, gestire e rimuovere tag per un repository. AWS CLI CodeCommit APIs Oltre a identificare, organizzare e tracciare il repository con i tag, puoi utilizzare i tag nelle policy IAM per controllare chi può visualizzare e interagire con il tuo repository. Per esempi di policy di accesso basate su tag, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Argomenti

- [Aggiunta di un tag a un repository](#)
- [Visualizzazione dei tag per un repository](#)
- [Modifica dei tag per un repository](#)

- [Rimuovere un tag da un repository](#)

Aggiunta di un tag a un repository

L'aggiunta di tag a un repository può aiutarti a identificare e organizzare le tue AWS risorse e a gestirne l'accesso. In primo luogo, è possibile aggiungere uno o più tag (coppie chiave-valore) a un repository. Il numero di tag che è possibile avere in un repository non ha limiti. Esistono limitazioni per i caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Limiti](#). Dopo aver creato i tag, puoi creare policy IAM per gestire l'accesso al repository in base a questi tag. Puoi usare la CodeCommit console o aggiungere tag AWS CLI a un repository.

Important

L'aggiunta di tag a un repository può avere impatto sull'accesso a tale repository. Prima di aggiungere un tag a un repository, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare i tag per controllare l'accesso a risorse come i repository. Per esempi di policy di accesso basate su tag, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Per ulteriori informazioni sull'aggiunta di tag a un repository al momento della creazione, consulta [Crea un repository \(console\)](#).

Argomenti

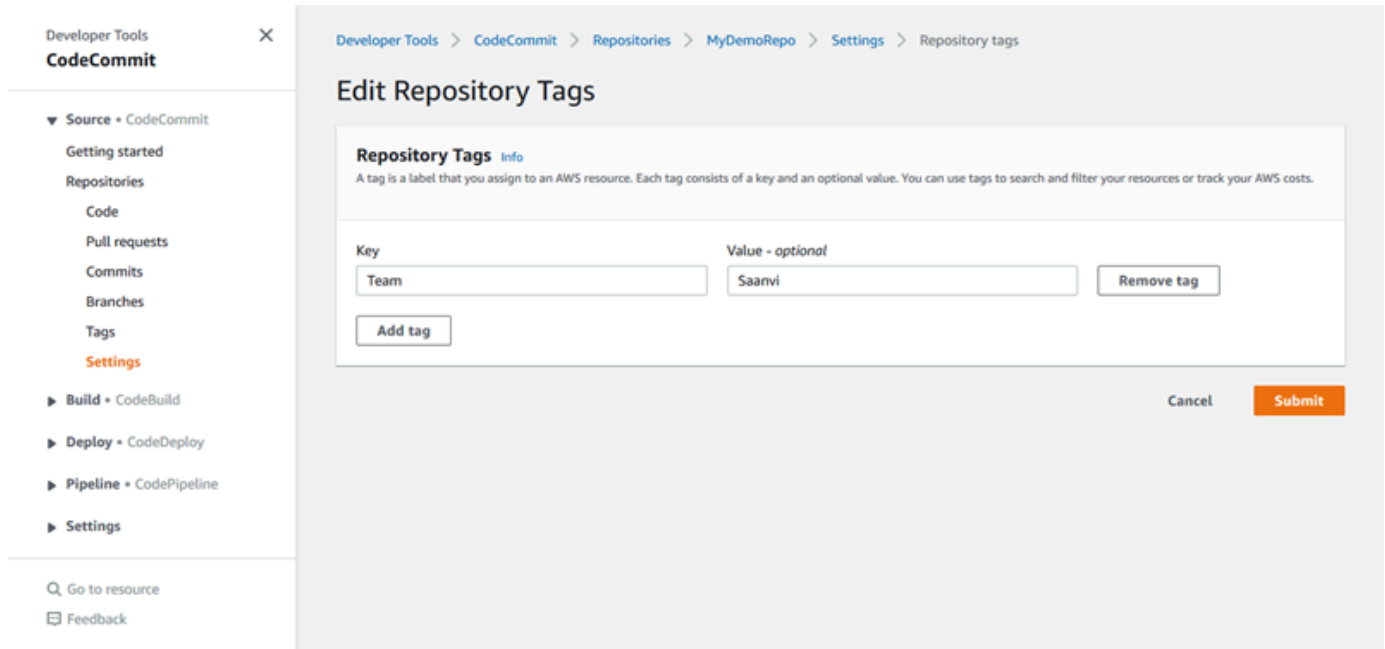
- [Aggiungi un tag a un repository \(console\)](#)
- [Aggiungi un tag a un repository \(\)AWS CLI](#)

Aggiungi un tag a un repository (console)

È possibile utilizzare la CodeCommit console per aggiungere uno o più tag a un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera aggiungere i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Repository tag (Tag repository).

- Se non sono stati aggiunti tag al repository, scegliere Add tag (Aggiungi tag). In caso contrario, scegliere Edit (Modifica), quindi scegliere Add tag (Aggiungi tag).
- Per Key (Chiave), inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).



- (Facoltativo) Per aggiungere un altro tag, scegliere Add tag (Aggiungi tag) .
- Una volta completata l'aggiunta di tag, scegliere Submit (Inoltra).

Aggiungi un tag a un repository (AWS CLI)

Segui questi passaggi per utilizzare AWS CLI per aggiungere un tag a un CodeCommit repository. Per aggiungere un tag a un repository al momento della creazione, consulta [Crea un repository \(AWS CLI\)](#).

In queste fasi, si assume che sia già installata una versione recente della AWS CLI o che sia aggiornata alla versione corrente. Per ulteriori informazioni, consultare [Installing the AWS Command Line Interface](#).

Al terminale o alla riga di comando, eseguire il comando `tag-resource`, specificando l'ARN (Amazon Resource Name) del repository in cui aggiungere i tag e la chiave e il valore del tag che desideri aggiungere. È possibile aggiungere più di un tag a un repository. Ad esempio, per etichettare un repository denominato *MyDemoRepo* con due tag, una chiave di tag denominata *Status* con il valore del tag e una chiave di *Secret* tag denominata *Team* con il valore del tag di: *Saanvi*

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Status=Secret,Team=Saanvi
```

In caso di successo, questo comando non restituisce alcun risultato.

Visualizzazione dei tag per un repository

I tag possono aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. Per ulteriori informazioni sulle strategie di tagging, consulta [Tagging AWS Resources](#). Per esempi di policy di accesso basate su tag, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Visualizza i tag per un repository (console)

È possibile utilizzare la CodeCommit console per visualizzare i tag associati a un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera visualizzare i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Repository tag (Tag repository).

The screenshot shows the AWS CodeCommit console interface. On the left is a navigation sidebar with a 'Developer Tools' header and a 'CodeCommit' section. Under 'CodeCommit', there are links for 'Getting started', 'Repositories', 'Code', 'Pull requests', 'Commits', 'Branches', 'Tags', 'Settings' (highlighted in orange), 'Build • CodeBuild', 'Deploy • CodeDeploy', 'Pipeline • CodePipeline', and 'Settings'. The main content area has a breadcrumb trail: 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Settings'. Below this is the title 'MyDemoRepo' and a tabbed interface with 'Settings', 'Notifications', 'Triggers', and 'Repository tags' (selected and highlighted in orange). The 'Repository tags' tab displays a table titled 'Repository Tags' with an 'Info' link and an 'Edit' button. The table has two columns: 'Key' and 'Value'. It contains two rows: 'Status' with value 'Secret' and 'Team' with value 'Saanvi'. There is a pagination control showing '1' and navigation arrows.

Key	Value
Status	Secret
Team	Saanvi

Visualizza i tag per un repository ()AWS CLI

Segui questi passaggi per utilizzare AWS CLI per visualizzare i AWS tag di un CodeCommit repository. Se non sono stati aggiunti tag, l'elenco restituito è vuoto.

Dal terminale o dalla riga di comando, esegui il comando `list-tags-for-resource`. Ad esempio, per visualizzare un elenco di valori di chiavi e di tag per un repository denominato *MyDemoRepo* con l'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
```

Se il comando viene eseguito correttamente, restituisce informazioni simili alle seguenti:

```
{  
  "tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Modifica dei tag per un repository

È possibile modificare il valore di un tag associato a un repository. Puoi anche cambiare il nome della chiave, il che equivale a rimuovere il tag corrente e aggiungerne uno diverso con il nuovo nome e lo stesso valore dell'altra chiave. Tieni presente che esistono limitazioni sul numero di caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Limiti](#).

Important

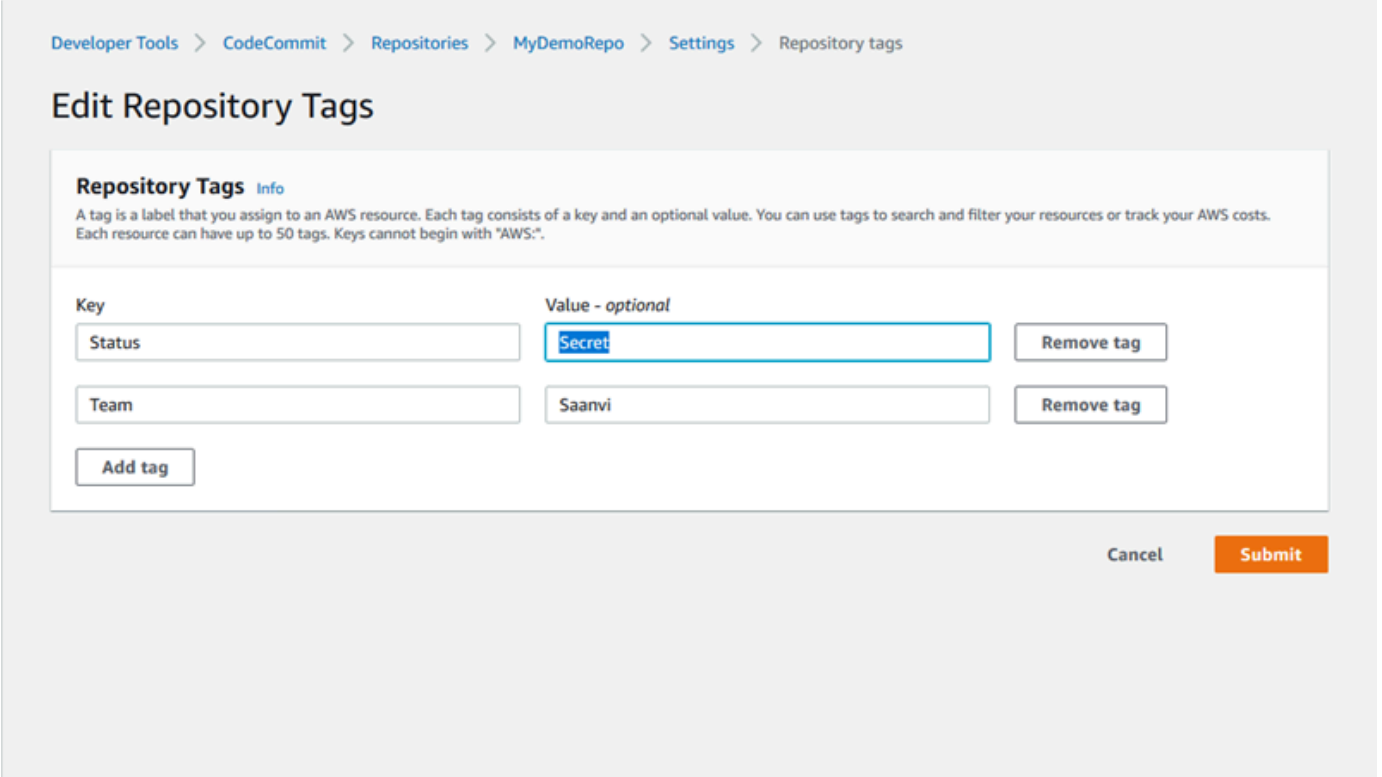
La modifica di tag a un repository può avere impatto sull'accesso a tale repository. Prima di modificare il nome (chiave) o il valore di un tag per un repository, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come i repository. Per esempi di policy di accesso basate su tag, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Modifica un tag per un repository (console)

È possibile utilizzare la CodeCommit console per modificare i tag associati a un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera modificare i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Repository tag (Tag repository).
4. Scegli Modifica.

5.



The screenshot shows the 'Edit Repository Tags' page in the AWS CodeCommit console. The breadcrumb trail at the top reads: 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Settings > Repository tags'. The main heading is 'Edit Repository Tags'. Below it, there is a section titled 'Repository Tags' with an 'Info' icon. The text explains: '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:".' The interface contains two columns: 'Key' and 'Value - optional'. Under 'Key', there are two text input fields: the first contains 'Status' and the second contains 'Team'. Under 'Value - optional', there are two text input fields: the first contains 'Secret' and the second contains 'Saanvi'. To the right of each 'Value' field is a 'Remove tag' button. At the bottom left of the form area is an 'Add tag' button. At the bottom right of the page are 'Cancel' and 'Submit' buttons.

Esegui una delle seguenti operazioni:

- Per modificare il tag, immettere un nuovo nome in Key (Chiave). La modifica del nome del tag equivale alla sostituzione del tag esistente con un nuovo tag che ha un nuovo nome chiave.
 - Per modificare il valore di un tag, immettere un nuovo valore. Se si desidera modificare il valore in un valore vuoto, eliminare il valore corrente e lasciare il campo vuoto.
6. Al termine delle modifiche dei tag, scegliere Submit (Inoltra).

Modifica i tag per un repository (AWS CLI)

Segui questi passaggi per utilizzare AWS CLI per aggiornare un tag per un CodeCommit repository. È possibile modificare il valore di una chiave esistente o aggiungere un'altra chiave.

Al terminale o nella riga di comando, esegui il comando `tag-resource` specificando l'ARN (Amazon Resource Name) del repository in cui desideri aggiornare un tag e specificare la chiave e il valore di tag:

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Team=Li
```

Rimuovere un tag da un repository

È possibile rimuovere uno o più tag associati a un repository. La rimozione di un tag non elimina il tag dalle altre AWS risorse associate a quel tag.

Important

La rimozione di tag da un repository può avere impatto sull'accesso a tale repository. Prima di rimuovere un tag da un repository, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come i repository. Per esempi di policy di accesso basate su tag, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Rimuovi un tag da un repository (console)

È possibile utilizzare la CodeCommit console per rimuovere l'associazione tra un tag e un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository da cui si desidera rimuovere i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Repository tag (Tag repository).
4. Scegli Modifica.
5. Cercare il tag da rimuovere, quindi scegliere Remove tag (Rimuovi tag).
6. Una volta terminata la rimozione di tag, scegliere Submit (Inoltra).

Rimuovi un tag da un repository (AWS CLI)

Segui questi passaggi per utilizzare il AWS CLI per rimuovere un tag da un CodeCommit repository. La rimozione di un tag non lo elimina completamente, ma rimuove semplicemente l'associazione tra il tag e il repository.

Note

Se elimini un CodeCommit repository, tutte le associazioni di tag vengono rimosse dal repository eliminato. Non è necessario rimuovere i tag prima di eliminare un repository.

Al terminale o nella riga di comando, eseguire il comando `untag-resource` specificando l'ARN del repository da cui desideri rimuovere i tag e la relativa chiave. Ad esempio, per rimuovere un tag su un repository denominato *MyDemoRepo* con la chiave tag: *Status*

```
aws codecommit untag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tag-keys Status
```

In caso di successo, questo comando non restituisce alcun risultato. Per verificare i tag associati al repository, eseguire il comando `list-tags-for-resource`.

Gestisci i trigger per un repository AWS CodeCommit

Puoi configurare un CodeCommit repository in modo che i push di codice o altri eventi attivino azioni, come l'invio di una notifica da Amazon Simple Notification Service (Amazon SNS) o l'invocazione di una funzione in AWS Lambda. Puoi creare fino a 10 trigger per ogni repository. CodeCommit

Generalmente, i trigger vengono configurati per:

- Inviare e-mail agli utenti abbonati ogni volta che qualcuno effettua il push nel repository.
- Inviare una notifica a un sistema di compilazione esterno per avviare una compilazione dopo che un utente effettua il push al ramo principale del repository.

Scenari come la notifica a un sistema di compilazione esterno richiedono la scrittura di una funzione Lambda per interagire con altre applicazioni. Lo scenario di posta elettronica richiede semplicemente la creazione di un argomento Amazon SNS.

Questo argomento mostra come impostare le autorizzazioni che consentono di CodeCommit attivare azioni in Amazon SNS e Lambda. Include inoltre collegamenti a esempi di creazione, modifica, test ed eliminazione di trigger.

Argomenti

- [Crea la risorsa e aggiungi le autorizzazioni per CodeCommit](#)
- [Esempio: creare un AWS CodeCommit trigger per un argomento di Amazon SNS](#)
- [Esempio: creare un AWS CodeCommit trigger per una funzione AWS Lambda](#)
- [Esempio: creare un trigger AWS CodeCommit per una AWS Lambda funzione esistente](#)
- [Modifica i trigger per un repository AWS CodeCommit](#)
- [Test dei trigger per un repository AWS CodeCommit](#)
- [Eliminare i trigger da un repository AWS CodeCommit](#)

Crea la risorsa e aggiungi le autorizzazioni per CodeCommit

Puoi integrare argomenti di Amazon SNS e funzioni Lambda con i trigger in CodeCommit, ma devi prima creare e poi configurare le risorse con una policy che conceda CodeCommit le autorizzazioni per interagire con tali risorse. È necessario creare la risorsa nello stesso luogo in cui si trova il repository. Regione AWS CodeCommit Ad esempio, se il repository si trova negli Stati Uniti orientali (Ohio) (us-east-2), l'argomento Amazon SNS o la funzione Lambda devono trovarsi negli Stati Uniti orientali (Ohio).

- Per gli argomenti di Amazon SNS, non è necessario configurare policy o autorizzazioni IAM aggiuntive se l'argomento Amazon SNS viene creato utilizzando lo stesso account del repository. CodeCommit Puoi creare il CodeCommit trigger non appena hai creato e sottoscritto l'argomento Amazon SNS.
 - Per ulteriori informazioni sulla creazione di argomenti in Amazon SNS, consulta la documentazione di [Amazon SNS](#).
 - Per informazioni sull'uso di Amazon SNS per inviare messaggi alle code di Amazon SQS, [consulta Sending Messages to Amazon SQS Queues nella Amazon SNS Developer Guide](#).
 - Per informazioni sull'uso di Amazon SNS per richiamare una funzione Lambda, consulta [Invoking Lambda Functions nella Amazon SNS Developer Guide](#).
- Puoi configurare le funzioni Lambda creando il trigger nella console Lambda come parte della funzione. Questo è il metodo più semplice, perché i trigger creati nella console Lambda includono

automaticamente le autorizzazioni necessarie CodeCommit per richiamare la funzione Lambda. Se si crea il trigger in CodeCommit, è necessario includere una policy che CodeCommit consenta di richiamare la funzione. Per ulteriori informazioni, consulta [Creare un trigger per una funzione Lambda esistente](#).

Esempio: creare un AWS CodeCommit trigger per un argomento di Amazon SNS

Puoi creare un trigger per un CodeCommit repository in modo che gli eventi in quel repository attivino le notifiche da un argomento di Amazon Simple Notification Service (Amazon SNS). Potresti voler creare un trigger per un argomento di Amazon SNS per consentire agli utenti di iscriversi alle notifiche sugli eventi del repository, come l'eliminazione delle filiali. Puoi anche sfruttare l'integrazione degli argomenti di Amazon SNS con altri servizi, come Amazon Simple Queue Service (Amazon SQS) e AWS Lambda

Note

- Devi indirizzare il trigger su un argomento esistente di Amazon SNS che è l'azione intrapresa in risposta agli eventi del repository. Per ulteriori informazioni sulla creazione e la sottoscrizione agli argomenti di Amazon SNS, [consulta la pagina Guida introduttiva ad Amazon Simple Notification Service](#).
- Gli argomenti FIFO (first in, first out) di Amazon SNS non sono supportati per i trigger CodeCommit

Argomenti

- [Crea un trigger per un argomento Amazon SNS per un CodeCommit repository \(console\)](#)
- [Crea un trigger per un argomento Amazon SNS per un CodeCommit repository \(AWS CLI\)](#)

Crea un trigger per un argomento Amazon SNS per un CodeCommit repository (console)

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repository, selezionare il repository in cui si desidera creare i trigger per gli eventi del repository.

3. Nel riquadro di navigazione per il repository, scegliere Settings (Impostazioni), quindi Triggers (Trigger).
4. Selezionare Create trigger (Crea trigger) e procedere come descritto di seguito:

- In Trigger name, inserisci un nome per il trigger (ad esempio,). *MyFirstTrigger*
- In Eventi, scegli gli eventi del repository che attivano l'argomento Amazon SNS per l'invio di notifiche.

Se si sceglie All repository events (Tutti gli eventi del repository), non è possibile scegliere altri eventi. Per scegliere un sottoinsieme di eventi, deselezionare All repository events (Tutti gli eventi del repository), quindi scegliere uno o più eventi nell'elenco. Se ad esempio si desidera eseguire il trigger solo quando un utente crea un ramo o un tag nel repository CodeCommit, deselezionare All repository events (Tutti gli eventi del repository), quindi scegliere Create branch or tag (Crea ramo o tag).

- Se si desidera che il trigger sia valido per tutti i rami del repository in Branches (Rami), lasciare la selezione vuota, poiché questa opzione di default applica il trigger automaticamente a tutti i rami. Se si desidera che questo trigger sia valido solo per rami specifici, scegliere fino a 10 nomi di rami dall'elenco dei rami del repository.
- In Scegli il servizio da usare, scegli Amazon SNS.
- In Amazon SNS, scegli il nome di un argomento dall'elenco o inserisci l'ARN per l'argomento.

Note

Gli argomenti FIFO (first in, first out) di Amazon SNS non sono supportati per i trigger. CodeCommit Devi scegliere un argomento di Amazon SNS il cui tipo sia impostato su Standard.

- In Dati personalizzati, fornisci tutte le informazioni facoltative che desideri includere nella notifica inviata dall'argomento Amazon SNS (ad esempio, il nome di un canale IRC utilizzato dagli sviluppatori quando discutono dello sviluppo in questo repository). Questo campo è una stringa e non può essere utilizzato per passare parametri dinamici.
5. (Facoltativo) Scegliere Test trigger (Esegui test trigger). Questo passaggio ti aiuta a confermare che l'accesso è stato configurato correttamente tra CodeCommit e l'argomento Amazon SNS. Utilizza l'argomento Amazon SNS per inviare una notifica di test utilizzando i dati del tuo

repository, se disponibili. Se non sono disponibili dati reali, la notifica di prova contiene dati di esempio.

6. Scegliere Create trigger (Crea trigger) per completare la creazione del trigger.

Crea un trigger per un argomento Amazon SNS per un CodeCommit repository ()AWS CLI

Puoi anche utilizzare la riga di comando per creare un trigger per un argomento Amazon SNS in risposta a eventi del CodeCommit repository, ad esempio quando qualcuno invia un commit al tuo repository.

Per creare un trigger per un argomento di Amazon SNS

1. Aprire un editor di testo semplice e creare un file JSON che specifichi:

- Il nome dell'argomento Amazon SNS.

Note

Gli argomenti FIFO (first in, first out) di Amazon SNS non sono supportati per i trigger. CodeCommit Devi scegliere un argomento di Amazon SNS il cui tipo sia impostato su Standard.

- Il repository e i rami che si desidera monitorare con questo trigger. Se non si specifica alcun ramo, il trigger viene applicato a tutti i rami nel repository.
- Gli eventi che attivano il trigger.

Salvare il file.

Ad esempio, per creare un trigger per un repository denominato *MyDemoRepo* che pubblica tutti gli eventi del repository su un argomento di Amazon SNS denominato *MySNSTopic* per due rami e: *main preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyFirstTrigger",
```

```

        "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
        "customData": "",
        "branches": [
            "main", "preprod"
        ],
        "events": [
            "all"
        ]
    }
}

```

È necessario specificare un blocco di trigger nel file JSON per ogni trigger di un repository. Per creare più trigger per il repository, includere più blocchi di trigger nel file JSON. Tenere presente che tutti i trigger creati in questo file si riferiscono al repository specificato. Non è possibile creare trigger per più repository in un singolo file JSON. Se, ad esempio, si desidera definire due trigger per un repository, è possibile creare un file JSON con due blocchi di trigger. Nell'esempio che segue non sono specificati rami per il secondo trigger, pertanto il trigger viene applicato a tutti i rami:

```

{
  "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"
      ]
    }
  ]
}

```

```
    ]  
  }  
]  
}
```

È possibile creare trigger per eventi specifici, ad esempio quando viene eseguito il push di un commit in un repository. I tipi di eventi includono:

- `all` per tutti gli eventi nel repository e nei rami specificati.
- `updateReference` quando il push dei commit viene eseguito nel repository e nei rami specificati.
- `createReference` quando viene creato un nuovo ramo o tag nel repository specificato.
- `deleteReference` quando viene eliminato un nuovo ramo o tag nel repository specificato.

Note

È possibile utilizzare più tipi di evento in un trigger. Se, tuttavia, si inserisce `all` non è possibile specificare altri eventi.

Per visualizzare l'elenco completo dei tipi di evento validi, dal terminale o dal prompt dei comandi, immettere `aws codecommit put-repository-triggers help`.

Inoltre, è possibile includere una stringa in `customData`, ad esempio il nome di un canale IRC utilizzato dagli sviluppatori quando discutono di sviluppo in questo repository. Questo campo è una stringa e non può essere utilizzato per passare parametri dinamici. Questa stringa viene aggiunta come attributo al codice CodeCommit JSON restituito in risposta al trigger.

2. (Facoltativo) Da un terminale o dal prompt dei comandi, eseguire il comando `test-repository-triggers`. Questo test utilizza dati di esempio dal repository (o genera dati di esempio se non sono disponibili dati) per inviare una notifica agli abbonati dell'argomento Amazon SNS. Ad esempio, quanto segue viene utilizzato per verificare che il codice JSON nel file trigger denominato *trigger.json* sia valido e che CodeCommit possa essere pubblicato sull'argomento Amazon SNS:

```
aws codecommit test-repository-triggers --cli-input-json file://trigger.json
```

Se il comando viene eseguito correttamente, restituisce informazioni simili alle seguenti:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- Da un terminale o dal prompt dei comandi, eseguire il comando `put-repository-triggers` per creare il trigger in CodeCommit. Ad esempio, per utilizzare un file JSON denominato `trigger.json` per creare il trigger:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Questo comando restituisce un [ID di configurazione](#) simile al seguente:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

- Per visualizzare la configurazione del trigger, eseguire il comando `get-repository-triggers` specificando il nome del repository:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Questo comando restituisce la struttura di tutti i trigger configurati per il repository, producendo un output simile al seguente:

```
{
  "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. Per verificare la funzionalità del trigger, creare ed effettuare il push di un commit nel repository in cui è stato configurato il trigger. Dovresti vedere una risposta dall'argomento Amazon SNS. Ad esempio, se hai configurato l'argomento Amazon SNS per inviare un'e-mail, dovresti vedere un'e-mail di Amazon SNS nell'account e-mail sottoscritto all'argomento.

Di seguito è riportato un esempio di output di un'e-mail inviata da Amazon SNS in risposta a un push a un CodeCommit repository:

```

{
  "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",
    }
  ]
}

```

```
]
}
```

Esempio: creare un AWS CodeCommit trigger per una funzione AWS Lambda

È possibile creare un trigger per un CodeCommit repository in modo che gli eventi nel repository richiama una funzione Lambda. In questo esempio, crei una funzione Lambda che restituisce l'URL utilizzato per clonare il repository in un log Amazon. CloudWatch

Argomenti

- [Creazione della funzione Lambda](#)
- [Visualizza il trigger per la funzione Lambda nel repository AWS CodeCommit](#)

Creazione della funzione Lambda

Quando usi la console Lambda per creare la funzione, puoi anche creare un CodeCommit trigger per la funzione Lambda. I passaggi seguenti includono una funzione Lambda di esempio. L'esempio è disponibile in due lingue: JavaScript e Python. La funzione restituisce l'URLs usato per clonare un repository in un registro. CloudWatch

Per creare una funzione Lambda utilizzando un blueprint Lambda

1. Accedi a Console di gestione AWS e apri la console all' AWS Lambda indirizzo. <https://console.aws.amazon.com/lambda/>
2. Nella pagina Funzioni Lambda, scegli Crea funzione. (Se non hai mai usato Lambda prima, scegli Inizia ora.)
3. Nella pagina Create function (Crea funzione), scegliere Author from scratch (Crea da zero). In Function Name, fornisci un nome per la funzione, ad esempio *MyLambdaFunctionforCodeCommit*. In Runtime, scegliere la lingua che si desidera utilizzare per scrivere la funzione, quindi scegliere Create function (Crea funzione).
4. Nella scheda Configuration (Configurazione) scegliere Add trigger (Aggiungi trigger).
5. Nella configurazione Trigger, scegli CodeCommit dall'elenco a discesa dei servizi.

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 X

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 X

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

- In Nome archivio, scegli il nome del repository in cui desideri configurare un trigger che utilizza la funzione Lambda in risposta agli eventi del repository.

- In Nome trigger, inserisci un nome per il trigger (ad esempio,). *MyLambdaFunctionTrigger*
- In Eventi, scegli gli eventi del repository che attivano la funzione Lambda. Se si sceglie All repository events (Tutti gli eventi del repository), non è possibile scegliere altri eventi. Se ad esempio si desidera scegliere un sottoinsieme di eventi, deselezionare All repository events (Tutti gli eventi del repository) e scegliere gli eventi desiderati nell'elenco. Ad esempio, se desideri che il trigger venga eseguito solo quando un utente crea un tag o un ramo nel AWS CodeCommit repository, rimuovi Tutti gli eventi del repository, quindi scegli Crea ramo o tag.
- Se si desidera applicare il trigger a tutti i rami del repository, in Branches (Rami), scegliere All Branches (Tutti i rami). In caso contrario, scegliere Specific branches (Rami specifici). Per impostazione predefinita, viene aggiunto il ramo predefinito per il repository. È possibile mantenere o eliminare questo ramo dall'elenco. Scegliere fino a 10 nomi di rami nell'elenco di rami del repository.
- (Facoltativo) In Dati personalizzati, inserisci le informazioni che desideri includere nella funzione Lambda (ad esempio, il nome del canale IRC usato dagli sviluppatori per discutere dello sviluppo nel repository). Questo campo è una stringa e non può essere utilizzato per passare parametri dinamici.

Scegliere Aggiungi.

6. Nella pagina Configuration (Configurazione) in Function Code (Codice funzione), per il tipo di immissione codice, scegliere Edit code inline (Modifica codice inline). In Runtime, scegliere Node.js. Se si desidera creare una funzione Python di esempio, scegliere Python.
7. In Code entry type (Tipo di voce del codice), scegliere Edit code inline (Modifica codice inline), quindi sostituire il codice hello world con uno dei due esempi riportati di seguito.

Per 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}`,
    );
  }
};

```

Per 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. Nella scheda Autorizzazioni, in Ruolo di esecuzione, scegli il ruolo per aprirlo nella console IAM. Modificare la policy collegata per aggiungere l'autorizzazione GetRepository per il repository che si desidera venga utilizzato dal trigger.

Visualizza il trigger per la funzione Lambda nel repository AWS CodeCommit

Dopo aver creato la funzione Lambda, è possibile visualizzare e testare il trigger in AWS CodeCommit. Il test del trigger esegue la funzione in risposta agli eventi del repository specificati.

Per visualizzare e testare il trigger per la funzione Lambda

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), scegliere il repository in cui si desidera visualizzare i trigger.
3. Nel riquadro di navigazione per il repository, scegliere Settings (Impostazioni), quindi Triggers (Trigger).
4. Esaminare l'elenco dei trigger per il repository. Dovresti vedere il trigger che hai creato nella console Lambda. Selezionarlo dall'elenco e quindi scegliere Test trigger (Esegui test trigger). Questa opzione tenta di invocare la funzione con i dati di esempio relativi al repository, incluso l'ID commit più recente. Se non esiste una cronologia dei commit, vengono generati valori di esempio costituiti da zero. Questo ti aiuta a confermare di aver configurato correttamente l'accesso tra AWS CodeCommit e la funzione Lambda.

5. Per verificare ulteriormente la funzionalità del trigger, creare ed effettuare il push di un commit nel repository in cui è stato configurato il trigger. Dovresti vedere una risposta dalla funzione Lambda nella scheda Monitoraggio per quella funzione nella console Lambda. Dalla scheda Monitoraggio, scegli Visualizza i log in CloudWatch. La CloudWatch console si apre in una nuova scheda e visualizza gli eventi relativi alla funzione. Selezionare dall'elenco il flusso di log che corrisponde al momento in cui è stato eseguito il push del commit. Dovrebbero essere visualizzati dati di evento simili ai seguenti:

```
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
```

Esempio: creare un trigger AWS CodeCommit per una AWS Lambda funzione esistente

Il modo più semplice per creare un trigger che richiami una funzione Lambda consiste nel creare quel trigger nella console Lambda. Questa integrazione integrata garantisce che CodeCommit disponga delle autorizzazioni necessarie per eseguire la funzione. Per aggiungere un trigger per una funzione Lambda esistente, vai alla console Lambda e scegli la funzione. Nella scheda Triggers (Trigger) per la funzione, seguire la procedura in Add trigger (Aggiungi trigger). Questi passaggi sono simili a quelli in [Creazione della funzione Lambda](#).

Puoi anche creare un trigger per una funzione Lambda in un CodeCommit repository. A tale scopo è necessario scegliere una funzione Lambda esistente da richiamare. Richiede inoltre la configurazione manuale delle autorizzazioni necessarie per CodeCommit eseguire la funzione.

Argomenti

- [Configura manualmente le autorizzazioni per consentire l'esecuzione CodeCommit di una funzione Lambda](#)
- [Creare un trigger per la funzione Lambda in un CodeCommit repository \(console\)](#)
- [Crea un trigger per una funzione Lambda per un CodeCommit repository \(AWS CLI\)](#)

Configura manualmente le autorizzazioni per consentire l'esecuzione CodeCommit di una funzione Lambda

Se crei un trigger CodeCommit che richiama una funzione Lambda, devi configurare manualmente le autorizzazioni che consentono di CodeCommit eseguire la funzione Lambda. Per evitare questa configurazione manuale, considera invece la possibilità di creare il trigger per la funzione nella console Lambda.

Per consentire CodeCommit l'esecuzione di una funzione Lambda

1. Apri un editor di testo semplice e crea un file JSON che specifichi il nome della funzione Lambda, i dettagli del CodeCommit repository e le azioni che desideri consentire in Lambda, in modo simile al seguente:

```
{
  "FunctionName": "MyCodeCommitFunction",
  "StatementId": "1",
  "Action": "lambda:InvokeFunction",
  "Principal": "codecommit.amazonaws.com",
  "SourceArn": "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo",
  "SourceAccount": "111122223333"
}
```

2. Salvate il file come file JSON con un nome facile da ricordare (ad esempio, .json).
AllowAccessfromMyDemoRepo
3. Utilizzando il file JSON appena creato, nel terminale (Linux, macOS o Unix) o nella riga di comando (Windows), esegui `aws lambda add-permissions` il comando per aggiungere un'autorizzazione alla politica delle risorse associata alla tua funzione Lambda:

```
aws lambda add-permission --cli-input-json file://AllowAccessfromMyDemoRepo.json
```

Questo comando restituisce la sintassi JSON dell'istruzione della policy appena aggiunta, producendo un output simile al seguente:

```
{
  "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\"}
}
```

Per ulteriori informazioni sulle politiche delle risorse per le funzioni Lambda, vedere [AddPermission](#) [The Pull/Push Event Models](#) nella AWS Lambda User Guide.

4. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
5. Nel riquadro di navigazione della Dashboard, scegli Ruoli e nell'elenco dei ruoli seleziona *lambda_basic_execution*.
6. Nella pagina di riepilogo del ruolo, scegliere la scheda Permissions (Autorizzazioni) e, in Inline Policies (Policy inline), scegliere Create Role Policy (Crea policy ruolo).
7. Nella pagina Set Permissions (Imposta autorizzazioni), scegliere Policy Generator (Generatore di policy), quindi Select (Seleziona).
8. Nella pagina Edit Permissions (Modifica autorizzazioni), procedere nel modo seguente:
 - In Effect (Effetto), scegliere Allow (Consenti).
 - In AWS Servizio, scegli AWS CodeCommit.
 - In Azioni, seleziona GetRepository.
 - In Amazon Resource Name (ARN), digitare il nome ARN del repository, ad esempio `arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo`.

Selezionare Add Statement (Aggiungi istruzione), quindi Next Step (Fase successiva).

9. Nella pagina Review Policy (Revisiona policy), scegliere Apply Policy (Applica policy).

La dichiarazione relativa alla policy dovrebbe essere simile all'esempio seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt11111111",
      "Effect": "Allow",
      "Action": [
```

```
        "codecommit:GetRepository"
      ],
      "Resource": [
        "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo"
      ]
    }
  ]
}
```

Creare un trigger per la funzione Lambda in un CodeCommit repository (console)

Dopo aver creato la funzione Lambda, è possibile creare un trigger CodeCommit che esegue la funzione in risposta agli eventi del repository specificati.

Note

Prima di poter testare o eseguire correttamente il trigger per l'esempio, è necessario configurare le politiche che consentono di CodeCommit richiamare la funzione e la funzione Lambda per ottenere informazioni sul repository. Per ulteriori informazioni, consulta [Per consentire CodeCommit l'esecuzione di una funzione Lambda](#).

Per creare un trigger per una funzione Lambda

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repository, selezionare il repository in cui si desidera creare i trigger per gli eventi del repository.
3. Nel riquadro di navigazione per il repository, scegliere Settings (Impostazioni), quindi Triggers (Trigger).
4. Selezionare Create trigger (Crea trigger).
5. In Create trigger (Crea trigger), eseguire quanto segue:
 - In Trigger name, inserisci un nome per il trigger (ad esempio,). *MyLambdaFunctionTrigger*
 - In Eventi, scegli gli eventi del repository che attivano la funzione Lambda.

Se si sceglie All repository events (Tutti gli eventi del repository), non è possibile scegliere altri eventi. Se ad esempio si desidera scegliere un sottoinsieme di eventi, deselezionare All

repository events (Tutti gli eventi del repository) e scegliere gli eventi desiderati nell'elenco.

Se ad esempio si desidera eseguire il trigger solo quando un utente crea un tag o un ramo nel repository CodeCommit, rimuovere All repository events (Tutti gli eventi del repository), quindi scegliere Create branch or tag (Crea ramo o tag).

- Se si desidera che il trigger sia valido per tutti i rami del repository in Branches (Rami), lasciare la selezione vuota, poiché questa opzione di default applica il trigger automaticamente a tutti i rami. Se si desidera che questo trigger sia valido solo per rami specifici, scegliere fino a 10 nomi di rami dall'elenco dei rami del repository.
 - In Choose the service to use (Scegli il servizio da utilizzare), scegliere AWS Lambda.
 - Nella funzione Lambda, scegli il nome della funzione dall'elenco o inserisci l'ARN per la funzione.
 - (Facoltativo) In Dati personalizzati, inserisci le informazioni che desideri includere nella funzione Lambda (ad esempio, il nome del canale IRC usato dagli sviluppatori per discutere dello sviluppo nel repository). Questo campo è una stringa e non può essere utilizzato per passare parametri dinamici.
6. (Facoltativo) Scegliere Test trigger (Esegui test trigger). Questa opzione tenta di invocare la funzione con i dati di esempio relativi al repository, incluso l'ID commit più recente. Se non esiste una cronologia dei commit, vengono generati valori di esempio costituiti da zero. Questo ti aiuta a confermare di aver configurato correttamente l'accesso tra CodeCommit e la funzione Lambda.
 7. Scegliere Create trigger (Crea trigger) per completare la creazione del trigger.
 8. Per verificare la funzionalità del trigger, creare ed effettuare il push di un commit nel repository in cui è stato configurato il trigger. Dovresti vedere una risposta dalla funzione Lambda nella scheda Monitoraggio per quella funzione nella console Lambda.

Crea un trigger per una funzione Lambda per un CodeCommit repository (AWS CLI)

Puoi anche usare la riga di comando per creare un trigger per una funzione Lambda in risposta agli eventi del CodeCommit repository, ad esempio quando qualcuno invia un commit al tuo repository.

Per creare un trigger per una funzione Lambda

1. Aprire un editor di testo semplice e creare un file JSON che specifichi:
 - Il nome della funzione Lambda.
 - Il repository e i rami che si desidera monitorare con questo trigger. Se non si specifica alcun ramo, il trigger viene applicato a tutti i rami nel repository.

- Gli eventi che attivano il trigger.

Salvare il file.

Ad esempio, se desideri creare un trigger per un repository denominato *MyDemoRepo* che pubblica tutti gli eventi del repository in una funzione Lambda denominata *MyCodeCommitFunction* per due rami e: *main* *preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyLambdaFunctionTrigger",
      "destinationArn": "arn:aws:lambda:us-east-1:111122223333:function:MyCodeCommitFunction",
      "customData": "",
      "branches": [
        "main", "preprod"
      ],
      "events": [
        "all"
      ]
    }
  ]
}
```

È necessario specificare un blocco di trigger nel file JSON per ogni trigger di un repository. Per creare più trigger per un repository, includere blocchi aggiuntivi nel file JSON. Tenere presente che tutti i trigger creati in questo file si riferiscono al repository specificato. Non è possibile creare trigger per più repository in un singolo file JSON. Se, ad esempio, si desidera definire due trigger per un repository, è possibile creare un file JSON con due blocchi di trigger. Nell'esempio che segue non sono specificati rami nel secondo blocco di trigger, pertanto il trigger viene applicato a tutti i rami:

```
{
  "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"
        ]
    }
]
}

```

È possibile creare trigger per eventi specifici, ad esempio quando viene eseguito il push di un commit in un repository. I tipi di eventi includono:

- `all` per tutti gli eventi nel repository e nei rami specificati.
- `updateReference` quando il push dei commit viene eseguito nel repository e nei rami specificati.
- `createReference` quando viene creato un nuovo ramo o tag nel repository specificato.
- `deleteReference` quando viene eliminato un nuovo ramo o tag nel repository specificato.

Note

È possibile utilizzare più tipi di evento in un trigger. Se, tuttavia, si inserisce `all` non è possibile specificare altri eventi.

Per visualizzare l'elenco completo dei tipi di evento validi, dal terminale o dal prompt dei comandi, immettere `aws codecommit put-repository-triggers help`.

Inoltre, è possibile includere una stringa in `customData`, ad esempio il nome di un canale IRC utilizzato dagli sviluppatori quando discutono di sviluppo in questo repository. Questo campo è una stringa e non può essere utilizzato per passare parametri dinamici. Questa stringa viene aggiunta come attributo al codice CodeCommit JSON restituito in risposta al trigger.

2. (Facoltativo) Da un terminale o dal prompt dei comandi, eseguire il comando `test-repository-triggers`. Ad esempio, quanto segue viene utilizzato per verificare che il file JSON denominato *trigger.json* sia valido e che CodeCommit possa attivare la funzione Lambda. Questo test utilizza i dati di esempio per attivare la funzione se non sono disponibili dati reali.

```
aws codecommit test-repository-triggers --cli-input-json file://trigger.json
```

Se il comando viene eseguito correttamente, restituisce informazioni simili alle seguenti:

```
{
  "successfulExecutions": [
    "MyLambdaFunctionTrigger"
  ],
  "failedExecutions": []
}
```

3. Da un terminale o dal prompt dei comandi, eseguire il comando `put-repository-triggers` per creare il trigger in CodeCommit. Ad esempio, per utilizzare un file JSON denominato *trigger.json* per creare il trigger:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Questo comando restituisce un ID di configurazione, simile al seguente:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

4. Per visualizzare la configurazione del trigger, eseguire il comando `get-repository-triggers` specificando il nome del repository:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Questo comando restituisce la struttura di tutti i trigger configurati per il repository, producendo un output simile al seguente:

```
{
  "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. Per verificare la funzionalità del trigger, creare ed effettuare il push di un commit nel repository in cui è stato configurato il trigger. Dovresti vedere una risposta dalla funzione Lambda nella scheda Monitoraggio per quella funzione nella console Lambda.

Modifica i trigger per un repository AWS CodeCommit

È possibile modificare i trigger che sono stati creati per un repository. CodeCommit così come gli eventi e i rami dei trigger, l'azione eseguita in risposta all'evento e altre impostazioni.

Argomenti

- [Modifica un trigger per un repository \(console\)](#)
- [Modifica un trigger per un repository \(\)AWS CLI](#)

Modifica un trigger per un repository (console)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repository, selezionare il repository in cui si desidera modificare un trigger per gli eventi del repository.
3. Nel riquadro di navigazione per il repository, scegliere Settings (Impostazioni), quindi Triggers (Trigger).
4. Dall'elenco di trigger per il repository, scegliere il trigger che si desidera modificare, quindi selezionare Edit (Modifica).
5. Apportare le modifiche desiderate al trigger, quindi scegliere Save (Salva).

Modifica un trigger per un repository (AWS CLI)

1. In un terminale (Linux, macOS o Unix) o dal prompt dei comandi (Windows), esegui il `get-repository-triggers` comando per creare un file JSON con la struttura di tutti i trigger configurati per il tuo repository. Ad esempio, per creare un file JSON denominato *MyTriggers.json* con la struttura di tutti i trigger configurati per un repository denominato: *MyDemoRepo*

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>MyTriggers.json
```

Questo comando non restituisce nulla, ma un file denominato *MyTriggers.json* viene creato nella directory in cui è stato eseguito il comando.

2. Modificare il file JSON in un editor di testo semplice e apportare modifiche al blocco di trigger del trigger che si desidera modificare. Sostituire la coppia `configurationId` con una coppia `repositoryName`. Salvare il file.

Ad esempio, se desideri modificare un trigger denominato *MyFirstTrigger* nel repository denominato in *MyDemoRepo* modo che si applichi a tutti i rami, sostituiscilo `configurationId` con `repositoryName` e rimuovi quello specificato `main` e `preprod` i rami in *red italic text*. Per impostazione predefinita, se non sono specificati rami, il trigger viene applicato a tutti i rami nel repository:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": [  
    {
```

```
    "destinationArn": "arn:aws:sns:us-east-2:111122223333:MyCodeCommitTopic",
    "branches": [
      "main",
      "preprod"
    ],
    "name": "MyFirstTrigger",
    "customData": "",
    "events": [
      "all"
    ]
  }
]
```

3. Dal terminale o dalla riga di comando, esegui il comando `put-repository-triggers`. Questo aggiorna tutti i trigger per il repository, incluse le modifiche apportate al trigger:

MyFirstTrigger

```
aws codecommit put-repository-triggers --repository-name MyDemoRepo
file:///MyTriggers.json
```

Questo comando restituisce un ID di configurazione, simile al seguente:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

Test dei trigger per un repository AWS CodeCommit

È possibile testare i trigger che sono stati creati per un repository. CodeCommit Il test comporta l'esecuzione del trigger con dati di esempio del repository, tra cui l'ID del commit più recente. Se non esiste una cronologia dei commit per il repository, vengono generati valori di esempio costituiti da zero. I trigger di test ti aiutano a confermare di aver configurato correttamente l'accesso tra CodeCommit e la destinazione del trigger, che si tratti di una AWS Lambda funzione o di una notifica di Amazon Simple Notification Service.

Argomenti

- [Prova un trigger per un repository \(console\)](#)

- [Prova un trigger per un repository \(AWS CLI\)](#)

Prova un trigger per un repository (console)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repository, selezionare il repository in cui si desidera eseguire il test di un trigger per gli eventi del repository.
3. Nel riquadro di navigazione per il repository, scegliere Settings (Impostazioni), quindi Triggers (Trigger).
4. Scegliere il trigger che si desidera testare, quindi selezionare Test trigger (Esegui test trigger). Viene visualizzato un messaggio di esito positivo o negativo. In caso di successo, dovresti vedere anche una risposta all'azione corrispondente dalla funzione Lambda o dall'argomento Amazon SNS.

Prova un trigger per un repository (AWS CLI)

1. In un terminale (Linux, macOS o Unix) o dal prompt dei comandi (Windows), esegui il `get-repository-triggers` comando per creare un file JSON con la struttura di tutti i trigger configurati per il tuo repository. Ad esempio, per creare un file JSON denominato *TestTrigger.json* con la struttura di tutti i trigger configurati per un repository denominato: `MyDemoRepo`

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>TestTrigger.json
```

Questo comando crea un file denominato *TestTriggers.json* nella directory in cui è stato eseguito il comando.

2. Modificare il file JSON in un editor di testo semplice e apportare le modifiche all'istruzione trigger. Sostituire la coppia `configurationId` con una coppia `repositoryName`. Salvare il file.

Ad esempio, se desideri testare un trigger denominato *MyFirstTrigger* nel repository denominato in *MyDemoRepo* modo che si applichi a tutti i rami, sostituiscilo `configurationId` con `repositoryName` e quindi salva un file simile al seguente come *TestTrigger.json*:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": [  
    {  
      "triggerName": "MyFirstTrigger",  
      "configurationId": "MyDemoRepo",  
      "branchFilter": "/*/*/*",  
      "eventTypes": "push",  
      "enabled": true  
    }  
  ]  
}
```



```
{
  "destinationArn": "arn:aws:sns:us-
east-2:111122223333:MyCodeCommitTopic",
  "branches": [
    "main",
    "preprod"
  ],
  "name": "MyFirstTrigger",
  "customData": "",
  "events": [
    "all"
  ]
}
```

3. Dal terminale o dalla riga di comando, esegui il comando `test-repository-triggers`. Questo aggiorna tutti i trigger per il repository, incluse le modifiche apportate al trigger: *MyFirstTrigger*

```
aws codecommit test-repository-triggers --cli-input-json file://TestTrigger.json
```

Questo comando restituisce una risposta simile alla seguente:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

Eliminare i trigger da un repository AWS CodeCommit

È possibile eliminare i trigger se non sono più utilizzati. Non puoi annullare l'eliminazione di un trigger, ma puoi creare nuovamente un trigger.

Note

Se hai configurato uno o più trigger per il tuo repository, l'eliminazione del repository non elimina gli argomenti di Amazon SNS o le funzioni Lambda che hai configurato come

destinazioni di tali trigger. Assicurati di eliminare tali risorse, anche se non sono più necessarie.

Argomenti

- [Elimina un trigger da un repository \(console\)](#)
- [Elimina un trigger da un repository \(\)AWS CLI](#)

Elimina un trigger da un repository (console)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository in cui si desidera eliminare i trigger per gli eventi del repository.
3. Nel riquadro di navigazione per il repository, selezionare Settings (Impostazioni). In Settings (Impostazioni), scegliere Triggers (Trigger).
4. Scegliere il trigger da eliminare dall'elenco di trigger, quindi selezionare Delete (Elimina).
5. Nella finestra di dialogo, digitare delete (elimina) per confermare.

Elimina un trigger da un repository ()AWS CLI

1. In un terminale (Linux, macOS o Unix) o dal prompt dei comandi (Windows), esegui il `get-repository-triggers` comando per creare un file JSON con la struttura di tutti i trigger configurati per il tuo repository. Ad esempio, per creare un file JSON denominato *MyTriggers.json* con la struttura di tutti i trigger configurati per un repository denominato: MyDemoRepo

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>MyTriggers.json
```

Questo comando crea un file denominato *MyTriggers.json* nella directory in cui è stato eseguito il comando.

2. Modificare il file JSON in un editor di testo semplice e rimuovere il blocco di trigger per il trigger che si desidera eliminare. Sostituire la coppia `configurationId` con una coppia `repositoryName`. Salvare il file.

Ad esempio, se si desidera rimuovere un trigger denominato *MyFirstTrigger* dal repository denominato *MyDemoRepo*, è necessario sostituirlo con `configurationId` e `repositoryName` e rimuovere l'istruzione in *red 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. Dal terminale o dalla riga di comando, esegui il comando `put-repository-triggers`. Questo aggiorna i trigger per il repository ed elimina il trigger: *MyFirstTrigger*

```
aws codecommit put-repository-triggers --repository-name MyDemoRepo
file:///MyTriggers.json
```

Questo comando restituisce un ID di configurazione, simile al seguente:

```
{
```

```
"configurationId": "0123456-I-AM-AN-EXAMPLE"  
}
```

Note

Per eliminare tutti i trigger per un repository denominato *MyDemoRepo*, il tuo file JSON potrebbe essere simile a questo:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": []  
}
```

Associare o dissociare un AWS CodeCommit repository con Amazon Reviewer CodeGuru

Amazon CodeGuru Reviewer è un servizio automatizzato di revisione del codice che utilizza l'analisi dei programmi e l'apprendimento automatico per rilevare problemi comuni e consigliare correzioni nel codice Java o Python. Puoi associare i repository nel tuo account Amazon Web Services a CodeGuru Reviewer. Quando lo fai, CodeGuru Reviewer crea un ruolo collegato al servizio che consente a CodeGuru Reviewer di analizzare il codice in tutte le pull request create dopo l'associazione.

Dopo aver associato un repository, CodeGuru Reviewer analizza e commenta eventuali problemi riscontrati durante la creazione di richieste pull. Ogni commento è chiaramente contrassegnato come proveniente da CodeGuru Reviewer con la denominazione Amazon CodeGuru Reviewer. Puoi rispondere a questi commenti proprio come faresti a qualsiasi altro commento in una richiesta pull e puoi anche fornire un feedback sulla qualità del suggerimento. Questo feedback viene condiviso con CodeGuru Reviewer e può aiutare a migliorare il servizio e i suoi suggerimenti.

Note

Non vedrai i commenti di CodeGuru Reviewer nelle pull request create prima che il repository fosse associato ad esso. È possibile che non vengano visualizzati commenti nelle richieste pull create dopo l'associazione per i seguenti motivi:

- La pull request non contiene codice Java o Python.

- CodeGuru Il revisore non ha avuto abbastanza tempo per eseguire e rivedere il codice nella pull request. Questo processo può richiedere fino a 30 minuti. I commenti possono apparire man mano che la revisione procede, ma i commenti non sono completi finché lo stato del lavoro non viene visualizzato come Completato.
- CodeGuru Il revisore non ha riscontrato alcun problema nel codice Java o Python nella pull request.
- Impossibile eseguire il processo di revisione del codice. Per verificare lo stato di una recensione per una pull request, consulta la scheda Attività della pull request.
- Stai visualizzando le modifiche alla pull request nella scheda Modifiche, la pull request è stata aggiornata e Amazon CodeGuru Reviewer non ha riscontrato alcun problema nelle modifiche. I commenti di Amazon CodeGuru Reviewer vengono visualizzati nella scheda Modifiche solo se sono stati inseriti nella revisione più recente della pull request. Vengono sempre visualizzati nella scheda Attività.

The screenshot displays the AWS CodeCommit pull request interface. At the top, the breadcrumb navigation shows 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 25'. The pull request title is '25: Updated some of our Java samples'. On the right, there are buttons for 'Approve', 'Close pull request', and 'Merge'. Below the title, a status bar indicates 'Open', 'No approval rules', 'No merge conflicts', 'Destination: main', 'Source: bugfix-1236', 'Author: Li_Juan', and 'Approvals: 0'. The 'Activity' tab is selected, showing the 'Amazon CodeGuru Reviewer job status' as 'In progress'. The 'Activity history' section shows a recent update: 'Pull request updated 1 minute ago. One or more commits added. Li_Juan updated the pull request.' Below this, a comment from 'Amazon CodeGuru Reviewer' is shown, stating: 'This code might not produce accurate results if the operation returns paginated results instead of all results. Consider adding another call to check for additional results. Leave feedback on this recommendation by selecting "Reply". Feedback and comments will also be shared with Amazon CodeGuru Reviewer and might be used to improve the service.' The comment is for 'line 100 of EventHandler.java' and includes a code snippet: 'ObjectListing files = s3Client.listObjects(bucketName);'. There is a 'Reply' button and a thumbs-up icon with a count of 1.

Per ulteriori informazioni [Lavorare con le richieste pull nei AWS CodeCommit repository](#) [Revisione di una richiesta di pull](#), consulta la [Guida per l'utente di Amazon CodeGuru Reviewer](#).

 Note

Devi aver effettuato l'accesso con un utente o un ruolo IAM che disponga di autorizzazioni sufficienti per associare o dissociare un repository con Reviewer. CodeGuru Per informazioni sulle politiche gestite CodeCommit che includono queste autorizzazioni, consulta e. [AWS politiche gestite per CodeCommit](#) [AWS CodeCommit politiche gestite e Amazon CodeGuru Reviewer](#) Per informazioni sulle autorizzazioni e sulla sicurezza di Amazon CodeGuru Reviewer, consulta la Amazon CodeGuru Reviewer User Guide.

Argomenti

- [Associa un repository a Reviewer CodeGuru](#)
- [Annulla l'associazione di un repository da Reviewer CodeGuru](#)

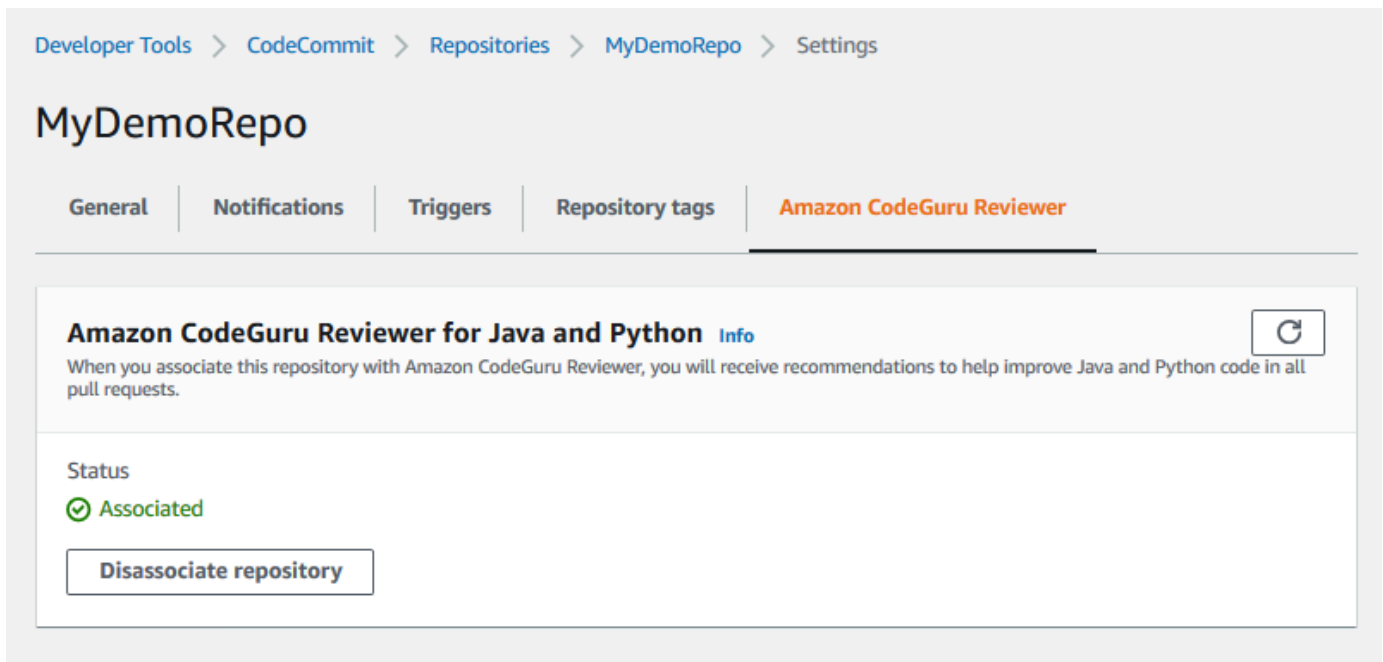
Associa un repository a Reviewer CodeGuru

Usa la AWS CodeCommit console per associare rapidamente un repository a Reviewer. CodeGuru Per altri metodi, consulta la Amazon CodeGuru Reviewer User Guide.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories, scegli il nome del repository da associare a Reviewer. CodeGuru
3. Scegli Impostazioni, quindi scegli Amazon CodeGuru Reviewer.
4. Scegliere Associate repository (Associa repository).

 Note

L'associazione completa di un repository a Reviewer può richiedere fino a 10 minuti. CodeGuru Lo stato non verrà aggiornato automaticamente. Per visualizzare lo stato corrente, scegliere il pulsante di aggiornamento.



Annulla l'associazione di un repository da Reviewer CodeGuru

Usa la AWS CodeCommit console per dissociare rapidamente un repository da Reviewer. CodeGuru. Per altri metodi, consulta la Amazon CodeGuru Reviewer User Guide.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories, scegli il nome del repository che desideri dissociare da Reviewer. CodeGuru.
3. Scegli Impostazioni, quindi scegli Amazon CodeGuru Reviewer.
4. Scegli Disassociate repository (Disassocia repository).

Visualizza i dettagli del CodeCommit repository

Puoi usare la AWS CodeCommit console o Git da un repository locale connesso al CodeCommit repository per visualizzare informazioni sui repository disponibili. AWS CLI

Prima di seguire le istruzioni, completa i passaggi descritti in [Configurazione](#).

Argomenti

- [Visualizza i dettagli del repository \(console\)](#)

- [Visualizza i dettagli CodeCommit del repository \(Git\)](#)
- [Visualizza i dettagli CodeCommit del repository \(\)AWS CLI](#)

Visualizza i dettagli del repository (console)

Usa la AWS CodeCommit console per visualizzare rapidamente tutti i repository creati con il tuo account Amazon Web Services.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories, visualizza i dettagli sui repository in Regione AWS cui hai effettuato l'accesso. Usa il selettore della regione per scegliere una Regione AWS diversa per visualizzare i repository in quella regione.
3. Scegli il nome del repository per il quale desideri visualizzare maggiori dettagli, quindi esegui una delle seguenti operazioni:
 - Per visualizzare l'URL per clonare il repository, scegliere Clone URL (Clona URL), quindi scegliere il protocollo che si desidera utilizzare durante la clonazione del repository. In questo modo si copia l'URL del clone. Per rivederlo, incollalo in un editor di testo normale.
 - Per visualizzare le opzioni configurabili per il repository e dettagli come l'ARN del repository e l'ID del repository, nel riquadro di navigazione, scegli Impostazioni.

Note

Se hai effettuato l'accesso come utente IAM, puoi configurare e salvare le tue preferenze per la visualizzazione del codice e altre impostazioni della console. Per ulteriori informazioni, consulta [Utilizzo delle preferenze dell'utente](#).

Visualizza i dettagli CodeCommit del repository (Git)

Per usare Git da un repository locale per visualizzare i dettagli sui CodeCommit repository, esegui il `git remote show` comando.

Prima di eseguire questi passaggi, collega il repository locale al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

1. Esegui il git remote show **remote-name** comando, **remote-name** dov'è l'alias del CodeCommit repository (per impostazione predefinita,). origin

 Tip

Per ottenere un elenco dei nomi dei CodeCommit repository e relativi URLs, esegui il comando. git remote -v

Ad esempio, per visualizzare i dettagli sul CodeCommit repository con l'alias: origin

```
git remote show origin
```

2. Per 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)
```

Per 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)
```

i Tip

Per cercare l'ID della chiave SSH per il tuo utente IAM, apri la console IAM ed espandi Security Credentials nella pagina dei dettagli utente IAM. L'ID della chiave SSH è disponibile in SSH Keys for. AWS CodeCommit

Per altre opzioni, consulta la documentazione di Git.

Visualizza i dettagli CodeCommit del repository ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per utilizzare il comando AWS CLI per visualizzare i dettagli del repository, esegui i seguenti comandi:

- [Per visualizzare un elenco di nomi di CodeCommit repository e i relativi nomi IDs, esegui list-repositories.](#)
- [Per visualizzare informazioni su un singolo CodeCommit repository, esegui get-repository.](#)
- Per visualizzare informazioni su più repository in, esegui. CodeCommit [batch-get-repositories](#)

Per visualizzare un elenco di repository CodeCommit

1. Eseguire il comando list-repositories:

```
aws codecommit list-repositories
```

Puoi utilizzare le opzioni `--sort-by` o `--order` per modificare l'ordine delle informazioni restituite.

2. In caso di successo, questo comando genera un `repositories` oggetto che contiene i nomi e IDs di tutti i repository CodeCommit associati all'account Amazon Web Services.

Questo è un esempio di output in base al comando precedente:

```
{
```

```
"repositories": [  
  {  
    "repositoryName": "MyDemoRepo",  
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"  
  },  
  {  
    "repositoryName": "MyOtherDemoRepo",  
    "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"  
  }  
]  
}
```

Per visualizzare i dettagli su un singolo repository CodeCommit

1. Esegui il `get-repository` comando, specificando il nome del CodeCommit repository con l'opzione. `--repository-name`

 Tip

[Per ottenere il nome del repository, esegui il comando CodeCommit `list-repositories`.](#)

Ad esempio, per visualizzare i dettagli su un repository denominato: CodeCommit MyDemoRepo

```
aws codecommit get-repository --repository-name MyDemoRepo
```

2. In caso di esito positivo, l'output del comando è un oggetto `repositoryMetadata` con le informazioni riportate di seguito:
 - Il nome del repository (`repositoryName`).
 - La descrizione del repository (`repositoryDescription`).
 - L'ID univoco del repository, generato dal sistema (`repositoryId`).
 - L'ID dell'account Amazon Web Services associato al repository (`accountId`).

Questo è un esempio di output in base al comando di esempio precedente:

```
{  
  "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"  
  }  
}
```

Per visualizzare i dettagli su più repository CodeCommit

1. Esegui il comando `batch-get-repositories` con l'opzione `--repository-names`. Aggiungi uno spazio tra il nome di ogni CodeCommit repository.

Tip

Per inserire i nomi dei repository CodeCommit, esegui il comando [list-repositories](#).

Ad esempio, per visualizzare i dettagli su due CodeCommit repository denominati e:
`MyDemoRepo` `MyOtherDemoRepo`

```
aws codecommit batch-get-repositories --repository-names MyDemoRepo MyOtherDemoRepo
```

2. In caso di successo, l'output del comando è un oggetto con le informazioni riportate di seguito.
 - Un elenco di tutti i CodeCommit repository che non sono stati trovati (`()repositoriesNotFound`).
 - Un elenco di CodeCommit repository (`()repositories`). Il nome di ogni CodeCommit repository è seguito da:
 - La descrizione del repository (`repositoryDescription`).
 - L'ID univoco del repository, generato dal sistema (`repositoryId`).

- L'ID dell'account Amazon Web Services associato al repository (accountId).

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "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": []
}
```

Modifica le impostazioni del AWS CodeCommit repository

È possibile utilizzare la AWS CodeCommit console AWS CLI e per modificare le impostazioni di un CodeCommit repository, ad esempio la descrizione o il nome.

Important

Modificare il nome di un repository può causare problemi a eventuali repository locali che utilizzano il nome precedente nell'URL remoto. Eseguire il comando `git remote set-url` per aggiornare l'URL remoto e utilizzare il nuovo nome di repository.

Argomenti

- [Modifica le impostazioni del repository \(console\)](#)
- [Modifica le impostazioni AWS CodeCommit del repository \(\)AWS CLI](#)

Modifica le impostazioni del repository (console)

Per utilizzare la AWS CodeCommit console per modificare le impostazioni di un CodeCommit repository AWS CodeCommit, segui questi passaggi.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository per cui si desidera modificare le impostazioni.
3. Nel pannello di navigazione scegli Impostazioni.
4. Per modificare il nome del repository, in Repository name (Nome repository), immettere un nuovo nome nella casella di testo Name (Nome) e scegliere Save (Salva). Quando richiesto, confermare la scelta.

Important

La modifica del nome del AWS CodeCommit repository cambierà l'SSH e l'HTTPS di URLS cui gli utenti hanno bisogno per connettersi al repository. Gli utenti non riusciranno a connettersi a questo repository finché non aggiorneranno le impostazioni di connessione. Inoltre, poiché l'ARN del repository cambierà, la modifica del nome del repository invaliderà le policy dell'utente IAM basate su tale ARN.

Per connettersi al repository dopo la modifica del nome, ogni utente deve utilizzare il comando `git remote set-url` e specificare il nuovo URL da impiegare. Ad esempio, se hai cambiato il nome del repository da `MyDemoRepo` a `MyRenamedDemoRepo`, gli utenti che utilizzano HTTPS per connettersi al repository eseguiranno il seguente comando Git:

```
git remote set-url origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Gli utenti che utilizzano SSH per connettersi al repository dovranno eseguire il seguente comando Git:

```
git remote set-url origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Per altre opzioni, consulta la documentazione di Git.

5. Per modificare la descrizione del repository, modificare il testo nella casella di testo Description (Descrizione) e quindi scegliere Save (Salva).

Note

Il campo della descrizione mostra Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza GetRepository o BatchGetRepositories APIs e prevedi di visualizzare il campo di descrizione del repository in un browser web, consulta l'CodeCommit API Reference.](#)

6. Per modificare il ramo predefinito, in Default branch (Ramo predefinito), scegliere prima l'elenco a discesa e poi un altro ramo. Scegli Save (Salva).
7. Per modificare la chiave di AWS KMS crittografia utilizzata per crittografare e decrittografare i dati nel repository, in Chiave di crittografia del repository, scegli Chiave gestita da AWS o Chiave gestita dal cliente per specificare il tipo di chiave da utilizzare. Se scegli una chiave gestita dal cliente, inserisci l'ARN della chiave. Scegli Save (Salva).
8. Per eliminare il repository, scegliere Delete repository (Elimina repository). Nella casella accanto a Type the name of the repository to confirm deletion (Digita il nome del repository per confermare l'eliminazione), immettere **delete** e quindi scegliere Delete (Elimina).

 Important

Dopo aver eliminato questo repository AWS CodeCommit, non sarà più possibile clonarlo su alcun repository locale o condiviso. Inoltre, non sarà più possibile estrarre dati da esso o inviargli dati da alcun repository locale o condiviso. Questa operazione non può essere annullata.

Modifica le impostazioni AWS CodeCommit del repository ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Da utilizzare AWS CLI per modificare le impostazioni di un CodeCommit repository in AWS CodeCommit, esegui uno o più dei seguenti comandi:

- [update-repository-description](#) per modificare la descrizione di un CodeCommit repository.
- [update-repository-name](#) per cambiare il nome di un CodeCommit repository.

Per modificare la descrizione di un CodeCommit repository

1. Eseguire il comando `update-repository-description` specificando:

- Il nome del CodeCommit repository (con l'opzione `--repository-name`).

 Tip

Per ottenere il nome del CodeCommit repository, esegui il [list-repositories](#) comando.

- La nuova descrizione del repository (con l'opzione `--repository-description`).

 Note

Il campo della descrizione mostra Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza GetRepository o BatchGetRepositories APIs e prevedi di visualizzare il](#)

[campo di descrizione del repository in un browser web, consulta l'CodeCommit API Reference.](#)

Ad esempio, per modificare la descrizione del CodeCommit repository denominato MyDemoRepo in: This description was changed

```
aws codecommit update-repository-description --repository-name MyDemoRepo --
repository-description "This description was changed"
```

Questo comando restituisce un output solo se ci sono errori.

2. Per verificare la descrizione modificata, esegui il get-repository comando, specificando il nome del CodeCommit repository di cui hai modificato la descrizione con l'opzione. --repository-name

L'output del comando mostra il testo modificato in repositoryDescription.

Per modificare il nome di un CodeCommit repository

1. Eseguire il comando update-repository-name specificando:
 - Il nome corrente del CodeCommit repository (con l'--old-nameopzione).

Tip

Per ottenere il nome del CodeCommit repository, esegui il comando [list-repositories](#).

- Il nuovo nome del CodeCommit repository (con l'opzione). --new-name

Ad esempio, per modificare il repository denominato MyDemoRepo in MyRenamedDemoRepo:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-name
MyRenamedDemoRepo
```

Questo comando restituisce un output solo se ci sono errori.

 Important

La modifica del nome del AWS CodeCommit repository modifica gli SSH e HTTPS URLs necessari agli utenti per connettersi al repository. Gli utenti non riusciranno a connettersi a questo repository finché non aggiorneranno le impostazioni di connessione. Inoltre, poiché l'ARN del repository cambia, la modifica del nome del repository invalida tutte le policy utente IAM che si basano sull'ARN di questo repository.

2. Per verificare il nome modificato, eseguire il comando `list-repositories` e rivedere l'elenco dei nomi di repository.

Sincronizza le modifiche tra un repository locale e un repository AWS CodeCommit

Si usa Git per sincronizzare le modifiche tra un repository locale e il CodeCommit repository connesso al repository locale.

Per inviare le modifiche dal repository locale al repository, esegui. CodeCommit `git push remote-name branch-name`

Per estrarre le modifiche al repository locale dal CodeCommit repository, esegui. `git pull remote-name branch-name`

Sia per spingere che per tirare, *remote-name* è il soprannome che il repository locale usa per il repository. CodeCommit *branch-name* è il nome del ramo del CodeCommit repository su cui effettuare il push o da cui estrarre.

 Tip

Per ottenere il soprannome che il repository locale usa per il CodeCommit repository, esegui. `git remote` Per ottenere un elenco di nomi di rami, eseguire `git branch`. Accanto al ramo corrente viene visualizzato un asterisco (*). È inoltre possibile eseguire `git status` per visualizzare il nome del ramo corrente.

 Note

Se hai clonato il repository, dal punto di vista del repository locale, non *remote-name* è il nome del repository. CodeCommit Quando si clona un repository, viene impostato automaticamente su. *remote-name* origin

Ad esempio, per inviare le modifiche dal repository locale al main ramo del CodeCommit repository con il soprannome: origin

```
git push origin main
```

Analogamente, per estrarre le modifiche al repository locale dal main ramo del CodeCommit repository con il soprannome: origin

```
git pull origin main
```

 Tip

Se si aggiunge l'opzione -u a git push, si impostano informazioni di registrazione upstream. Ad esempio, se corrigi `git push -u origin main`), in futuro puoi correre `git push` e `git pull` senza *remote-namebranch-name*. Per ottenere le informazioni di registrazione upstream, eseguire `git remote show remote-name` (ad esempio, `git remote show origin`).

Per altre opzioni, consulta la documentazione di Git.

Invia i commit a un repository Git aggiuntivo

È possibile configurare il repository locale per eseguire il push delle modifiche a due repository remoti. Ad esempio, potrebbe essere necessario continuare a utilizzare la soluzione di repository Git esistente per sperimentare AWS CodeCommit. Segui questi passaggi di base per inviare le modifiche nel tuo repository locale a CodeCommit un repository Git separato.

 Tip

Se non disponi di un repository Git, puoi crearne uno vuoto su un servizio diverso da CodeCommit e quindi migrare il tuo CodeCommit repository su di esso. È consigliabile seguire i passaggi simili a quelli indicati in [Migrare a CodeCommit](#).

1. Dal prompt dei comandi o da un terminale, passare alla directory del repository locale ed eseguire il comando `git remote -v`. Verrà visualizzato un output simile al seguente:

Per 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)
```

Per 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. Esegui il `git remote set-url --add --push origin git-repository-name` comando where **git-repository-name** è l'URL e il nome del repository Git in cui desideri ospitare il codice. Questo modifica la destinazione push di origin a quel repository Git.

 Note

`git remote set-url --add --push` sostituisce l'URL predefinito per i push, quindi è necessario eseguire due volte il comando, come dimostrato nelle fasi successive.

Ad esempio, il comando seguente modifica il push of origin in **some-URL**/MyDestinationRepo:

```
git remote set-url --add --push origin some-URL/MyDestinationRepo
```

Questo comando non restituisce alcun risultato.

 Tip

Se stai eseguendo il push verso un repository Git che richiede credenziali, assicurati di configurare tali credenziali in un helper per le credenziali o nella configurazione della stringa. *some-URL* in caso contrario, i push a tale repository non riescono.

3. Eseguire di nuovo il comando `git remote -v`, che dovrebbe creare un output simile al seguente:

Per HTTPS:

```
origin  https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  some-URL/MyDestinationRepo (push)
```

Per SSH:

```
origin  ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  some-URL/MyDestinationRepo (push)
```

4. Ora aggiungi il repository. CodeCommit Eseguire di nuovo il comando `git remote set-url --add --push origin`, ma questa volta selezionando l'URL e il nome del repository CodeCommit.

Ad esempio, il comando seguente aggiunge il push of origin a `https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo`:

Per HTTPS:

```
git remote set-url --add --push origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Per SSH:

```
git remote set-url --add --push origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

Questo comando non restituisce alcun risultato.

5. Eseguire di nuovo il comando `git remote -v`, che dovrebbe creare un output simile al seguente:

Per 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)
```

Per 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)
```

Ora hai due repository Git come destinazione per i tuoi push, ma i push vanno prima a *some-URL* /. MyDestinationRepo. Se il push a quel repository ha esito negativo, i commit non sono eseguiti a entrambi i repository.

 Tip

Se l'altro repository richiede credenziali che desideri inserire manualmente, valuta la possibilità di modificare l'ordine dei push in modo da inviare per primo. CodeCommit
Eseguire `git remote set-url --delete` per eliminare il primo repository verso il quale è stato effettuato il push, quindi eseguire `git remote set-url --add` per aggiungerlo nuovamente in modo che diventi la seconda destinazione di push nell'elenco.

Per altre opzioni, consulta la documentazione di Git.

6. Per verificare che i push siano effettuati verso entrambi i repository remoti, utilizzare un editor di testo per creare il seguente file di testo nel repository locale:

```
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. Eseguire `git add` per posizionare la modifica nel repository locale:

```
git add bees.txt
```

8. Eseguire `git commit` per eseguire il commit della modifica nel repository locale:

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

9. Per inviare il commit dal repository locale ai tuoi repository remoti, esegui `git push -u remote-name branch-name` where **remote-name** è il soprannome che il repository locale usa per i repository remoti ed **branch-name** è il nome del ramo da inviare al repository.

 Tip

È sufficiente utilizzare l'opzione `-u` al primo push. Vengono quindi impostate le informazioni di registrazione upstream.

Ad esempio, l'esecuzione di `git push -u origin main` dimostrerebbe l'esecuzione del push verso entrambi i repository remoti nei rami previsti, con un output simile a:

Per 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
```

Per 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
```

Per altre opzioni, consulta la documentazione di Git.

Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli

Puoi configurare l'accesso ai CodeCommit repository per utenti e gruppi IAM in un altro AWS account. Tale configurazione viene spesso definita accesso multiaccount. Questa sezione fornisce esempi e step-by-step istruzioni per configurare l'accesso tra account a un repository denominato *MySharedDemoRepo* nella regione Stati Uniti orientali (Ohio) in un AWS account (denominato AccountA) per utenti IAM che appartengono a un gruppo IAM denominato *DevelopersWithCrossAccountRepositoryAccess* in un altro account AWS (denominato AccountB).

La sezione si articola in tre parti:

- Operazioni dell'amministratore nell'AccountA.
- Operazioni dell'amministratore nell'AccountB.
- Operazioni dell'utente del repository nell'AccountB.

Per configurare l'accesso multiaccount:

- L'amministratore in AccountA accede come utente IAM con le autorizzazioni necessarie per creare e gestire repository e creare ruoli in CodeCommit IAM. Se utilizzi policy gestite, applica IAMFull Access e AWSCode CommitFullAccess a questo utente IAM.

L'ID account di esempio per l'AccountA è *111122223333*.

- L'amministratore di AccountB accede come utente IAM con le autorizzazioni necessarie per creare e gestire utenti e gruppi IAM e per configurare le policy per utenti e gruppi. Se utilizzi policy gestite, applica IAMFull Access a questo utente IAM.

L'ID account di esempio per l'AccountB è **888888888888**.

- L'utente del repository in AccountB, per emulare le attività di uno sviluppatore, accede come utente IAM membro del gruppo IAM creato per consentire l'accesso al CodeCommit repository in AccountA. Tale account deve essere configurato con:
 - AWS Accesso alla console di gestione.
 - Una chiave di accesso e una chiave segreta da utilizzare per la connessione alle AWS risorse e l'ARN del ruolo da assumere quando si accede ai repository in AccountA.
 - L'utilità git-remote-codecommit sul computer locale in cui viene clonato il repository. Questa utilità richiede Python e il programma di installazione (pip). È possibile scaricare l'utilità da [git-remote-codecommit](#) nel sito Web Python Package Index.

[Per ulteriori informazioni, consulta Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit e utenti IAM.](#)

Argomenti

- [Accesso all'archivio su più account: azioni per l'amministratore in AccountA](#)
- [Accesso all'archivio tra account: azioni per l'amministratore in AccountB](#)
- [Accesso all'archivio tra account: azioni per l'utente del repository in AccountB](#)

Accesso all'archivio su più account: azioni per l'amministratore in AccountA

Per autorizzare gli utenti o i gruppi nell'AccountB ad accedere a un repository nell'AccountA, un amministratore dell'AccountA deve:

- Creare una policy nell'AccountA che permette l'accesso al repository.
- Crea un ruolo in AccountA che possa essere assunto dagli utenti e dai gruppi IAM in AccountB.
- Collegare la policy al ruolo.

Le sezioni che seguono illustrano i passaggi e gli esempi.

Argomenti

- [Fase 1: Creare una politica per l'accesso al repository in AccountA](#)
- [Fase 2: Creare un ruolo per l'accesso al repository in AccountA](#)

Fase 1: Creare una politica per l'accesso al repository in AccountA

È possibile creare una policy in AccountA che conceda l'accesso al repository in AccountA agli utenti in AccountB. A seconda del livello di accesso che vuoi autorizzare, procedi in uno dei seguenti modi:

- Configurare la policy in modo da permettere agli utenti dell'AccountB di accedere a un repository specifico, ma non di visualizzare un elenco di tutti i repository nell'AccountA.
- Configurare un livello di accesso aggiuntivo per permettere agli utenti dell'AccountB di scegliere il repository da un elenco di tutti i repository nell'AccountA.

Per creare una policy per l'accesso al repository

1. Accedi alla console di AWS gestione come utente IAM con le autorizzazioni per creare policy in AccountA.
2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
3. Nel riquadro di navigazione, scegli Policy.
4. Scegli Create Policy (Crea policy).
5. Scegliere la scheda JSON e incollare il documento della seguente policy di JSON nella casella di testo JSON. Sostituisci *us-east-2* con Regione AWS for the repository, *111122223333* con l'ID account per AccountA *MySharedDemoRepo* e con il nome del CodeCommit tuo repository in 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"
    ]
}
]
}
```

Se desideri che gli utenti che assumono questo ruolo possano visualizzare un elenco di repository nella home page della CodeCommit console, aggiungi una dichiarazione aggiuntiva alla politica, come segue:

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": "*"
}
]
}

```

Questo tipo di accesso aiuta gli utenti che assumono questo ruolo con questa policy a individuare il repository a cui possono accedere. Possono scegliere il nome del repository dall'elenco ed essere indirizzati alla home page del repository condiviso (Code). Gli utenti non possono accedere a nessun altro repository di quelli visualizzati nell'elenco, ma possono visualizzare i repository nell'AccountA nella pagina Dashboard (Pannello di controllo).

Se non desideri consentire agli utenti che assumono il ruolo di visualizzare un elenco di tutti gli archivi in AccountA, utilizza il primo esempio di policy, ma assicurati di inviare a tali utenti un link diretto alla home page dell'archivio condiviso nella console. CodeCommit

6. Scegliere Esamina policy. Il validatore delle policy segnala errori di sintassi (ad esempio, se dimentichi di sostituire l'ID account Amazon Web Services di esempio e il nome del repository con l'ID dell'account Amazon Web Services e il nome del repository).
7. Nella pagina Rivedi la policy, inserisci un nome per la policy (ad esempio,). *CrossAccountAccessForMySharedDemoRepo* È anche possibile fornire una descrizione della policy. Scegli Crea policy.

Fase 2: Creare un ruolo per l'accesso al repository in AccountA

Dopo aver configurato una policy, crea un ruolo che gli utenti e i gruppi IAM in AccountB possono assumere e associa la policy a quel ruolo.

Per creare un ruolo per l'accesso al repository

1. Nella console IAM, scegliere Roles (Ruoli).

2. Scegli Crea ruolo.
3. Scegli un altro account Amazon Web Services.
4. In Account ID, inserisci l'ID dell'account Amazon Web Services per AccountB (ad esempio, **888888888888**). Scegli Successivo: autorizzazioni.
5. In Allega le politiche di autorizzazione, seleziona la politica creata nella procedura precedente (***CrossAccountAccessForMySharedDemoRepo***). Scegli Prossimo: Rivedi.
6. In Nome ruolo, inserisci un nome per il ruolo (ad esempio, ***MyCrossAccountRepositoryContributorRole***). È anche possibile immettere una descrizione del ruolo per aiutare gli altri utenti a capirne lo scopo.
7. Scegli Crea ruolo.
8. Aprire il ruolo appena creato e copiare l'ARN del ruolo (ad esempio ***arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole***). Questo ARN va inviato all'amministratore dell'AccountB.

Accesso all'archivio tra account: azioni per l'amministratore in AccountB

Per autorizzare gli utenti o i gruppi nell'AccountB ad accedere a un repository nell'AccountB, l'amministratore dell'AccountB deve creare un gruppo nell'AccountB. Tale gruppo deve essere configurato con una policy che consente ai membri del gruppo di assumere il ruolo creato dall'amministratore dell'AccountA.

Le sezioni che seguono illustrano i passaggi e gli esempi.

Argomenti

- [Fase 1: Creare un gruppo IAM per l'accesso al repository per gli utenti AccountB](#)
- [Fase 2: Creare una policy e aggiungere utenti al gruppo IAM](#)

Fase 1: Creare un gruppo IAM per l'accesso al repository per gli utenti AccountB

Il modo più semplice per gestire quali utenti IAM in AccountB possono accedere all'archivio AccountA consiste nel creare un gruppo IAM in AccountB autorizzato ad assumere il ruolo in AccountA e quindi aggiungere gli utenti IAM a quel gruppo.

Per creare un gruppo per l'accesso multiaccount al repository

1. Accedi alla console di AWS gestione come utente IAM con le autorizzazioni necessarie per creare gruppi e policy IAM e gestire gli utenti IAM in AccountB.
2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
3. Nella console IAM, scegli Gruppi.
4. Selezionare Create New Group (Crea nuovo gruppo).
5. In Nome gruppo, inserisci un nome per il gruppo (ad esempio, *DevelopersWithCrossAccountRepositoryAccess*). Selezionare Next Step (Fase successiva).
6. In Attach Policy (Collega policy), seleziona Next Step (Fase successiva); Nella procedura successiva verrà creata la policy multiaccount. Terminare la creazione del gruppo.

Fase 2: Creare una policy e aggiungere utenti al gruppo IAM

Ora che hai creato il gruppo, crea la policy che consente ai suoi membri di assumere il ruolo che dà loro accesso al repository nell'AccountA. Quindi aggiungi al gruppo gli utenti IAM in AccountB a cui desideri consentire l'accesso in AccountA.

Per creare una policy e aggiungere utenti al gruppo

1. Nella console IAM, scegli Gruppi, quindi scegli il nome del gruppo che hai appena creato (ad esempio, *DevelopersWithCrossAccountRepositoryAccess*).
2. Scegli la scheda Autorizzazioni. Espandere Inline Policies (Policy inline), quindi scegliere il collegamento per creare una policy inline. (Se si sta configurando un gruppo già provvisto di una policy inline, scegliere Create Group Policy (Crea policy di gruppo)).
3. Scegliere Custom Policy (Policy personalizzata) quindi Select (Seleziona).
4. In Policy Name (Nome policy), inserisci un nome per la policy, ad esempio *AccessPolicyForSharedRepository*.
5. In Policy Document (Documento policy), incollare la seguente policy. In **Resource**, sostituisci l'ARN con l'ARN della politica creata dall'amministratore in AccountA (ad esempio, `arn:aws:iam: ::role/`), quindi scegli Applica politica. **111122223333**
MyCrossAccountRepositoryContributorRole Per ulteriori informazioni sulla policy creata dall'amministratore nell'AccountA, consulta [Fase 1: Creare una politica per l'accesso al repository in AccountA](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource":
      "arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole"
  }
}
```

6. Scegli la scheda Users (Utenti); Scegli Aggiungi utenti al gruppo, quindi aggiungi gli utenti IAM AccountB. Ad esempio, potresti aggiungere un utente IAM con il nome utente *Saanvi_Sarkar* al gruppo.

Note

Gli utenti in AccountB devono disporre dell'accesso programmatico, inclusa una chiave di accesso e una chiave segreta, per configurare i computer locali per l'accesso all'archivio condiviso. CodeCommit Se stai creando utenti IAM, assicurati di salvare la chiave di accesso e la chiave segreta. Per garantire la sicurezza dell'account AWS , la chiave di accesso segreta è accessibile solo al momento della creazione.

Accesso all'archivio tra account: azioni per l'utente del repository in AccountB

Per accedere al repository nell'AccountA, gli utenti del gruppo AccountB devono configurare i loro computer locali per l'accesso al repository. Le sezioni che seguono illustrano i passaggi e gli esempi.

Argomenti

- [Fase 1: Configurare AWS CLI e Git per consentire a un utente AccountB di accedere al repository in AccountA](#)
- [Fase 2: Clona e accedi al CodeCommit repository in AccountA](#)

Fase 1: Configurare AWS CLI e Git per consentire a un utente AccountB di accedere al repository in AccountA

Non è possibile utilizzare chiavi SSH o credenziali Git per accedere ai repository in un altro account Amazon Web Services. Gli utenti AccountB devono configurare i propri computer per utilizzare git-remote-codecommit (consigliato) o l'helper delle credenziali per accedere all'archivio CodeCommit condiviso in AccountA. Puoi comunque continuare a utilizzare le chiavi SSH o le credenziali Git per accedere ai repository nell'AccountB.

Attieniti alla seguente procedura per configurare l'accesso utilizzando git-remote-codecommit. Se non l'hai già installato git-remote-codecommit, scaricalo dal sito Web [git-remote-codecommit](#) Python Package Index.

Per configurare AWS CLI e Git per l'accesso tra più account

1. Installa il AWS CLI sul computer locale. Consulta le istruzioni per il tuo sistema operativo nella sezione relativa all'[installazione di AWS CLI](#).
2. Installare Git sul computer locale. Per installare Git, ti suggeriamo siti Web quali [Git Downloads](#) o [Git for Windows](#).

Note

CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git. Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

3. Dal terminale o dalla riga di comando, nella posizione della directory dove si desidera clonare il repository, eseguire i comandi git config --local user.name e git config --local user.email per impostare il nome utente e la e-mail dei commit che si vuole apportare al repository. Esempio:

```
git config --local user.name "Saanvi Sarkar"  
git config --local user.email saanvi_sarkar@example.com
```

Questi comandi non restituiscono nulla, ma l'e-mail e il nome utente che hai specificato saranno associati ai commit apportati al repository nell'AccountA.

4. Eseguire il comando `aws configure --profile` per configurare un profilo di default da utilizzare durante la connessione alle risorse nell'AccountB. Quando richiesto, fornisci la chiave di accesso e la chiave segreta per il tuo utente IAM.

 Note

Se hai già installato AWS CLI e configurato un profilo, puoi saltare questo passaggio.

Ad esempio, esegui il comando seguente per creare un AWS CLI profilo predefinito da utilizzare per accedere alle AWS risorse in AccountB negli Stati Uniti orientali (Ohio) (us-east-2):

```
aws configure
```

Quando viene richiesto, fornire le seguenti informazioni:

```
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. Eseguire di nuovo il comando `aws configure --profile` per configurare un profilo con nome da utilizzare durante la connessione al repository nell'AccountA. Quando richiesto, fornisci la chiave di accesso e la chiave segreta per il tuo utente IAM. Ad esempio, esegui il comando seguente per creare un AWS CLI profilo denominato *MyCrossAccountAccessProfile* da utilizzare per accedere a un repository in AccountA negli Stati Uniti orientali (Ohio) (us-east-2):

```
aws configure --profile MyCrossAccountAccessProfile
```

Quando viene richiesto, fornire le seguenti informazioni:

```
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. In un editor di testo semplice, aprire il file `config`, anche chiamato file di configurazione di AWS CLI. A seconda del sistema operativo, questo file potrebbe trovarsi `~/.aws/config` in Linux, macOS o Unix oppure in *drive*:\Users\ *USERNAME* \config su Windows.

7. Nel file, trovare la voce corrispondente al profilo predefinito configurato per l'accesso ai repository nell'AccountB. La schermata visualizzata dovrebbe risultare simile a quella nell'immagine seguente:

```
[default]
region = us-east-2
output = json
```

Aggiungere account alla configurazione del profilo. Fornisci l'ID AWS account di AccountB.
Esempio:

```
[default]
account = 888888888888
region = us-east-2
output = json
```

8. Nel file, trova la voce che corrisponde al *MyCrossAccountAccessProfile* profilo appena creato. La schermata visualizzata dovrebbe risultare simile a quella nell'immagine seguente:

```
[profile MyCrossAccountAccessProfile]
region = us-east-2
output = json
```

Aggiungere account, role_arn e source_profile alla configurazione del profilo. Fornisci l'ID dell'account Amazon Web Services di accountA, l'ARN del ruolo in AccountA che assumi per accedere al repository nell'altro account e il nome del tuo profilo predefinito in AccountB. AWS CLI Esempio:

```
[profile MyCrossAccountAccessProfile]
region = us-east-2
account = 111122223333
role_arn = arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole
source_profile = default
output = json
```

Salvare le modifiche e chiudere l'editor di testo normale.

Fase 2: Clona e accedi al CodeCommit repository in AccountA

Eseguire git clone, git push e git pull per clonare il repository CodeCommit multiaccount ed eseguirne il push e l'estrazione. Puoi anche accedere alla console di AWS gestione, cambiare ruolo e utilizzare la CodeCommit console per interagire con l'archivio dell'altro account.

Note

A seconda di come è stato configurato il ruolo IAM, potresti essere in grado di visualizzare i repository nella pagina predefinita di CodeCommit. Se non riesci a visualizzare i repository, chiedi all'amministratore del repository di inviarti via email un link URL alla pagina Codici relativa all'archivio condiviso nella console. CodeCommit L'URL è simile al seguente:

```
https://console.aws.amazon.com/codecommit/home?region=us-east-2#/repository/MySharedDemoRepo/browse/HEAD/--/
```

Per clonare il repository multiaccount sul computer locale

1. Alla riga di comando o sul terminale, nella directory in cui si vuole clonare il repository, eseguire il comando git clone con l'URL clone HTTPS. Esempio:

```
git clone codecommit://MyCrossAccountAccessProfile@MySharedDemoRepo
```

Salvo diversa indicazione, il repository viene clonato in una sottodirectory con lo stesso nome del repository.

2. Modificare le directory nel repository clonato e aggiungere o apportare una modifica a un file. Ad esempio, puoi aggiungere un file denominato *NewFile.txt*
3. Aggiungi il file alle modifiche tracciate per il repository locale, esegui il commit della modifica e invia il file al repository. CodeCommit Esempio:

```
git add NewFile.txt  
git commit -m "Added a file to test cross-account access to this repository"  
git push
```

Per ulteriori informazioni, consulta [Guida introduttiva a Git e AWS CodeCommit](#).

Ora che hai aggiunto un file, vai alla CodeCommit console per visualizzare il tuo commit, rivedere le modifiche degli altri utenti al repository, partecipare alle pull request e altro ancora.

Per accedere all'archivio tra account nella console CodeCommit

1. Accedi Console di gestione AWS a accountB (**888888888888**) come utente IAM a cui è stato concesso l'accesso tra account diversi al repository in accountA.
2. Scegliere un nome utente nella barra di navigazione e nell'elenco a discesa scegliere Switch Role (Cambia ruolo).

 Note

Se è la prima volta che questa opzione viene selezionata, rivedere le informazioni nella pagina, quindi scegliere nuovamente Switch Role (Cambia ruolo).

3. Nella pagina Switch Role (Cambia ruolo), procedere come segue:
 - In Account, immettere l'ID account dell'AccountA (ad esempio **111122223333**).
 - In Ruolo, inserisci il nome del ruolo che desideri assumere per l'accesso al repository in AccountA (ad esempio **MyCrossAccountRepositoryContributorRole**).
 - In Display Name (Nome di visualizzazione), immettere un nome descrittivo del ruolo. Questo nome viene visualizzato nella console quando si assume il ruolo. Viene inoltre visualizzato nell'elenco dei ruoli assunti la prossima volta che si desidera cambiare ruoli nella console.
 - (Opzionale) In Color (Colore), scegliere un'etichetta colorata per il nome di visualizzazione.
 - Seleziona Switch Role (Cambia ruolo).

Per ulteriori informazioni, consulta [Passare a un ruolo \(Console di gestione AWS\)](#).

4. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).

Se il ruolo assunto dispone dell'autorizzazione per visualizzare i nomi dei repository nell'AccountA, comparirà un elenco dei repository e un messaggio di errore che informa delle mancate autorizzazioni a visualizzarne lo stato. Questo è il comportamento previsto. Scegliere dall'elenco il nome del repository condiviso.

Se il ruolo assunto non dispone dell'autorizzazione per visualizzare i nomi dei repository nell'AccountA, comparirà un messaggio di errore e un elenco vuoto senza repository. Incollare il

collegamento URL sul repository o modificare il collegamento alla console e cambiare `/list` nel nome del repository condiviso (ad esempio `/MySharedDemoRepo`).

5. In Code (Codice), trovare il nome del file aggiunto dal computer locale. Selezionare il nome per trovare il codice nel file, quindi esplorare il resto del repository e iniziare a utilizzarne le funzionalità.

Per ulteriori informazioni, consulta [Iniziare con AWS CodeCommit](#).

Elimina un repository AWS CodeCommit

È possibile utilizzare la CodeCommit console o AWS CLI eliminare un CodeCommit repository.

Note

Quando si elimina un repository le relative copie locali (repository locali) non vengono eliminate. Per eliminare un repository locale, utilizza gli strumenti di gestione delle directory e dei file del computer locale.

Argomenti

- [Eliminare un CodeCommit repository \(console\)](#)
- [Eliminare un repository locale](#)
- [Eliminare un CodeCommit repository \(AWS CLI\)](#)

Eliminare un CodeCommit repository (console)

Segui questi passaggi per utilizzare la CodeCommit console per eliminare un CodeCommit repository.

Important

Dopo aver eliminato un CodeCommit repository, non è più possibile clonarlo su alcun repository locale o condiviso. Inoltre, non è più possibile estrarre dati da esso o inviargli dati da alcun repository locale o condiviso. Questa operazione non può essere annullata.

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).

2. In Repositories (Repository) selezionare il nome del repository da eliminare.
3. Nel pannello di navigazione scegli Impostazioni.
4. Nella scheda General (Generale) in Delete repository (Elimina repository), scegliere Delete repository (Elimina repository). Immettere **delete**, quindi scegliere Delete (Elimina). Il repository viene eliminato definitivamente.

Note

L'eliminazione del repository in CodeCommit non elimina alcun repository locale.

Eliminare un repository locale

Utilizza gli strumenti di gestione delle directory e dei file del computer locale per eliminare la directory che contiene il repository locale.

L'eliminazione di un repository locale non elimina alcun CodeCommit repository a cui potrebbe essere connesso.

Eliminare un CodeCommit repository ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per utilizzare il AWS CLI per eliminare un CodeCommit repository, esegui il delete-repository comando, specificando il nome del CodeCommit repository da eliminare (con l'--repository-nameopzione).

Important

Dopo aver eliminato un CodeCommit repository, non è più possibile clonarlo su alcun repository locale o condiviso. Inoltre, non è più possibile estrarre dati da esso o inviargli dati da alcun repository locale o condiviso. Questa operazione non può essere annullata.

Tip

[Per ottenere il nome del repository, esegui il comando CodeCommit list-repositories.](#)

Ad esempio, per eliminare un repository denominato MyDemoRepo:

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

In caso di successo, l'ID del CodeCommit repository che è stato eliminato definitivamente viene visualizzato nell'output:

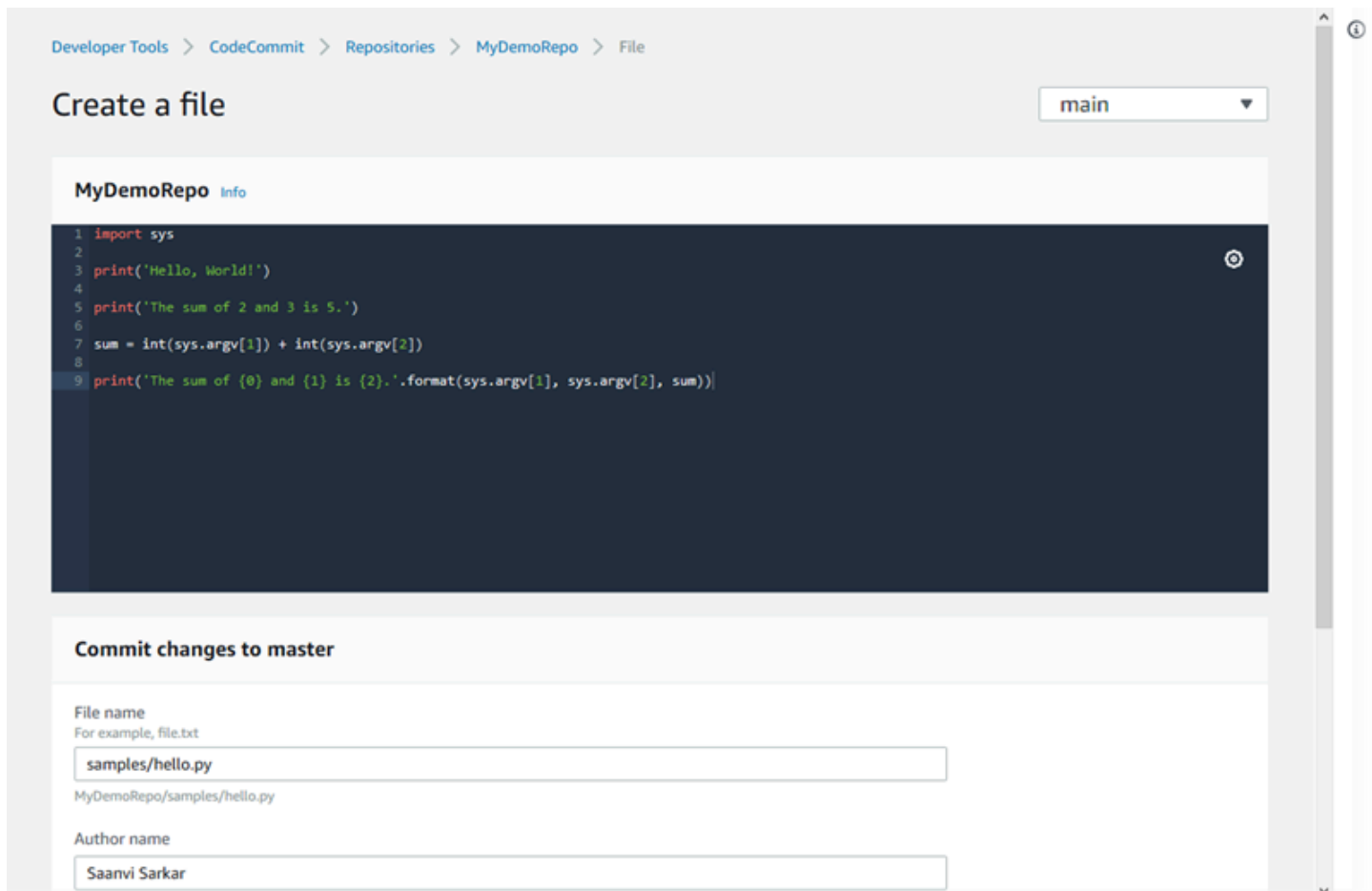
```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

L'eliminazione di un CodeCommit repository non elimina i repository locali che potrebbero essere collegati ad esso.

Lavorare con i file nei AWS CodeCommit repository

In CodeCommit, un file è un'informazione autonoma e controllata dalla versione a disposizione dell'utente e degli altri utenti del repository e del ramo in cui è archiviato il file. Puoi organizzare i file del repository con una struttura di directory, come su un computer. A differenza del computer, tiene traccia CodeCommit automaticamente di ogni modifica apportata a un file. Puoi confrontare le versioni di un file e memorizzare diverse versioni di un file in differenti rami del repository.

Per aggiungere o modificare un file in un repository, puoi utilizzare un client Git. Puoi anche utilizzare la CodeCommit console AWS CLI, l'API CodeCommit API.



Per informazioni sull'utilizzo di altri aspetti del repository in CodeCommit, consulta [Lavorare con i repository](#), [Utilizzo delle richieste pull](#), [Gestione dei rami](#), [Lavorare con i commit](#), e [Utilizzo delle preferenze dell'utente](#).

Argomenti

- [Sfogliare i file in un AWS CodeCommit repository](#)

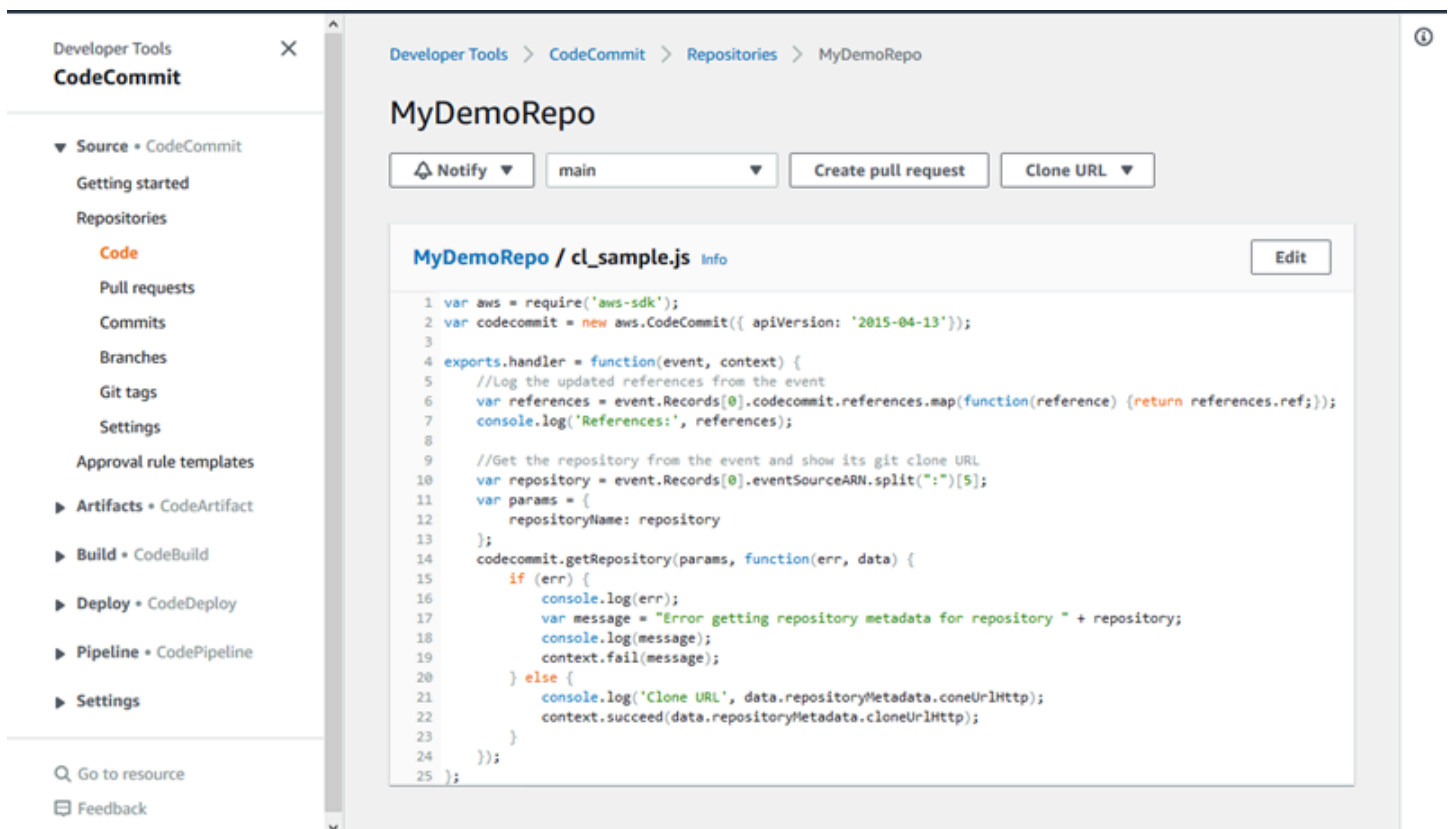
- [Creare o aggiungere un file a un AWS CodeCommit repository](#)
- [Modifica il contenuto di un file in un AWS CodeCommit repository](#)

Sfogliare i file in un AWS CodeCommit repository

Dopo esserti connesso a un CodeCommit repository, puoi clonarlo in un repository locale o utilizzare la CodeCommit console per sfoglierne il contenuto. Questo argomento descrive come utilizzare la CodeCommit console per sfogliare il contenuto di un repository. CodeCommit

Note

Per CodeCommit gli utenti attivi, non è previsto alcun costo per la navigazione del codice dalla CodeCommit console. Ulteriori informazioni sulle situazioni in cui possono verificarsi addebiti sono disponibili nella pagina relativa ai [prezzi](#).



Sfoggia un CodeCommit repository

È possibile utilizzare la CodeCommit console per esaminare i file contenuti in un repository o per leggere rapidamente il contenuto di un file.

Per esplorare il contenuto di un repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nella pagina Repositories (Repository), nell'elenco di repository, selezionare il repository che si desidera esplorare.
3. Nella visualizzazione Code (Codice), esplorare il contenuto del ramo predefinito per il repository.

Per modificare la visualizzazione su un altro ramo o tag, scegliere il pulsante di selezione della visualizzazione. Scegliere il nome di un ramo o tag dall'elenco a discesa oppure immettere il nome del ramo o del tag nella casella di filtro e quindi scegliere dall'elenco.

4. Esegui una di queste operazioni:
 - Per visualizzare il contenuto di una directory, sceglierla nell'elenco. È possibile scegliere una qualsiasi delle directory nell'elenco di navigazione per tornare alla visualizzazione di tale directory. È inoltre possibile utilizzare la freccia verso l'alto in cima all'elenco della directory.
 - Per visualizzare il contenuto di un file, sceglierlo nell'elenco. Se il file supera il limite di oggetti di commit, non potrà essere visualizzato nella console, ma solo in un repository locale. Per ulteriori informazioni, consulta [Quote](#). Per chiudere la visualizzazione dei file, scegliere la directory da visualizzare dalla barra di navigazione del codice.

Note

Non tutti i file binari sono visualizzabili nella console. Se si sceglie un file binario che è potenzialmente visualizzabile, viene visualizzato un messaggio di avviso che chiede di confermare che si desidera visualizzare il contenuto. Per visualizzare il file, scegliere Show file contents (Mostra contenuti file). Se non si desidera visualizzare il file, scegliere la directory da visualizzare dalla barra di navigazione del codice.

Se si sceglie un file markdown (.md), utilizzare i pulsanti Rendered Markdown (Markdown di rendering) e Markdown Source (Origine markdown) per passare alternativamente alla vista di sintassi e in rendering. Per ulteriori informazioni, consulta [Uso di Markdown nella console](#).

Creare o aggiungere un file a un AWS CodeCommit repository

Puoi usare la CodeCommit console o un client Git per aggiungere un file a un repository. AWS CLI
Puoi caricare un file dal computer locale al repository oppure utilizzare l'editor di codice della console per creare il file. L'editor è un modo rapido e semplice per aggiungere un file semplice, ad esempio un file readme.md, a un ramo in un repository.

Upload a file

MyDemoRepo [Info](#)

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

Commit changes to master

Author name

Maria Garcia

Email address

maria_garcia@example.com

Commit message - optional

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

Adding my first file to the repository.

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

Argomenti

- [Crea o carica un file \(console\)](#)
- [Aggiungi un file \(\)AWS CLI](#)
- [Aggiungi un file \(Git\)](#)

Crea o carica un file (console)

È possibile utilizzare la CodeCommit console per creare un file e aggiungerlo a un ramo in un CodeCommit repository. Durante la creazione dei file, puoi fornire il tuo nome utente e un indirizzo e-mail. Puoi anche aggiungere un messaggio di commit in modo che gli altri utenti sappiano chi ha aggiunto il file e perché. È anche possibile caricare un file direttamente dal tuo computer locale in un ramo di un repository.

Per aggiungere un file a un repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository in cui si desidera aggiungere un file.
3. Nella visualizzazione Code (Codice), selezionare il ramo in cui si desidera aggiungere il file. Per impostazione predefinita, quando si apre la visualizzazione Code (Codice) vengono visualizzati i contenuti del ramo predefinito.

Per modificare la visualizzazione su un altro ramo, scegliere il pulsante di selezione della visualizzazione. Scegliere il nome di un ramo dall'elenco a discesa oppure immettere il nome del ramo nella casella di filtro e quindi sceglierlo dall'elenco.

4. Selezionare Add file (Aggiungi file) e scegliere una delle seguenti opzioni:
 - Per utilizzare l'editor di codice per creare i contenuti di un file e aggiungerlo al repository, scegliere Create file (Crea file).
 - Per caricare un file dal computer locale nel repository, scegliere Upload file (Carica file).
5. Fornire le informazioni agli altri utenti su chi ha aggiunto questo file nel repository e perché.
 - In Author name (Nome creatore), immettere il proprio nome. Questo nome viene utilizzato sia come nome dell'autore che come nome del committer nelle informazioni di commit. CodeCommit utilizza per impostazione predefinita il nome utente IAM o una derivazione dell'accesso alla console come nome dell'autore.
 - In Indirizzo e-mail, inserisci un indirizzo e-mail in modo che altri utenti del repository possano contattarti in merito a questa modifica.
 - In Commit message (Messaggio di commit), immettere una breve descrizione. Questa fase è facoltativa, ma vivamente consigliata. In caso contrario, viene utilizzato un messaggio di commit predefinito.
6. Esegui una di queste operazioni:
 - Se si sta caricando un file, scegliere il file dal computer locale.
 - Se si crea un file, immettere il contenuto che si desidera aggiungere nell'editor di codice e fornire un nome per il file.
7. Scegliere Commit changes (Applica modifiche).

Aggiungi un file ()AWS CLI

È possibile utilizzare il AWS CLI `put-file` comando and per aggiungere un file in un CodeCommit repository. È inoltre possibile utilizzare il comando `put-file` per aggiungere un percorso o una struttura di directory per il file.

Note

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per aggiungere un file a un repository

1. Nel computer locale, creare il file che si desidera aggiungere al repository CodeCommit.
2. Dal terminale o dalla riga di comando, eseguire il comando `put-file`, specificando:
 - Il repository in cui si desidera aggiungere il file.
 - Il ramo in cui si desidera aggiungere il file.
 - L'ID commit completo dell'ultimo commit effettuato al ramo, noto anche come commit di estremità o testa.
 - Il percorso locale del file. La sintassi utilizzata per questo percorso varia a seconda del sistema operativo locale.
 - Il nome del file che si desidera aggiungere, inclusi il percorso in cui il file aggiornato è memorizzato nel repository, se presente.
 - Il nome utente e l'indirizzo e-mail che si desidera associare a questo file.
 - Un messaggio di commit che spiega perché è stato aggiunto il file.

Il nome utente, l'indirizzo e-mail e il messaggio di commit sono facoltativi, ma consentono agli altri utenti di sapere chi è l'autore e il motivo della modifica. Se non fornisci un nome utente, per CodeCommit impostazione predefinita utilizza il tuo nome utente IAM o una derivazione dell'accesso alla console come nome dell'autore.

Ad esempio, per aggiungere un file denominato *ExampleSolution.py* a un repository denominato *MyDemoRepo* a un ramo denominato *feature-randomizationfeature* il cui commit più recente ha un ID di: *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

Quando si aggiungono file binari, utilizzare `fileb://` per specificare il percorso locale del file.

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

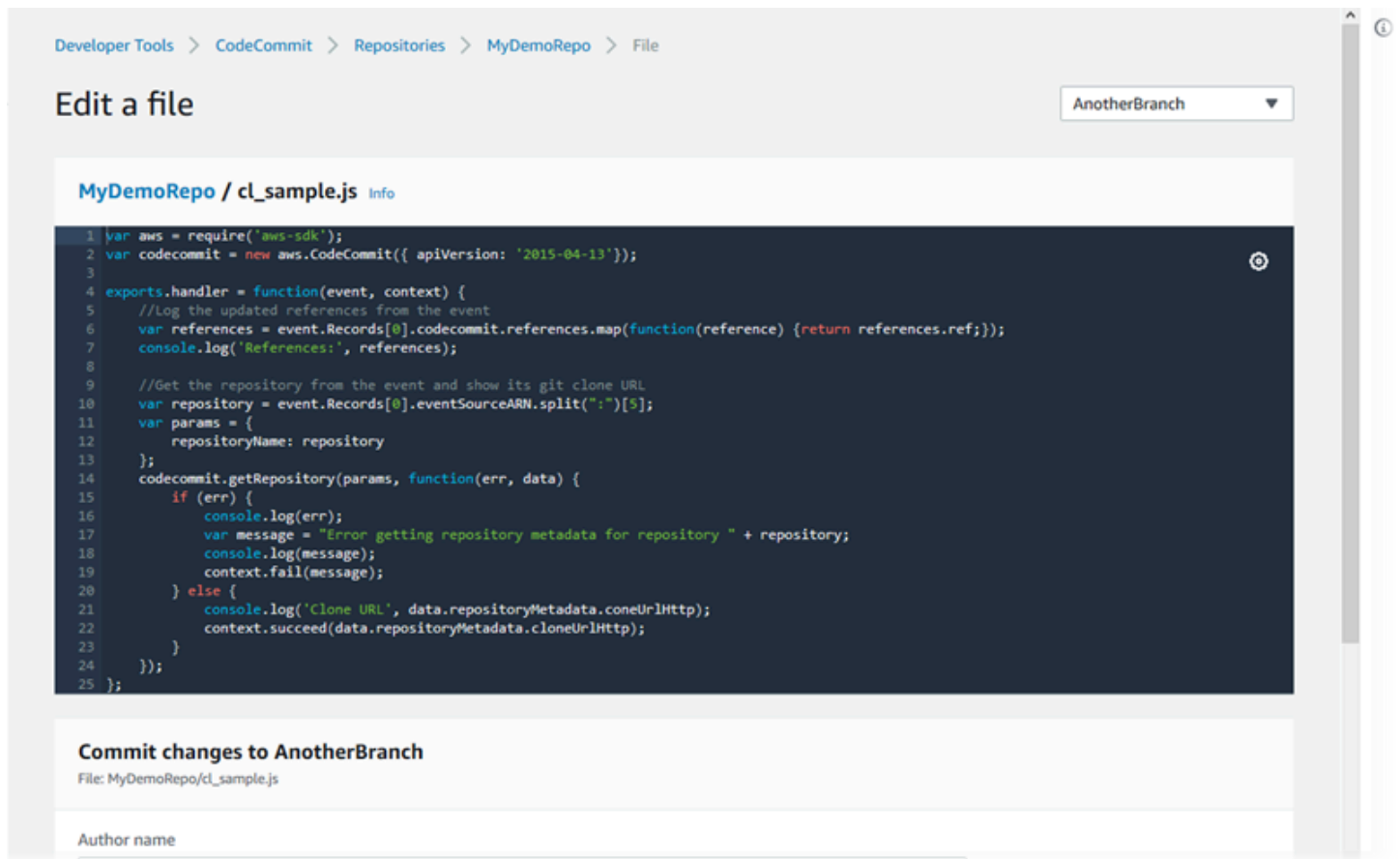
```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Aggiungi un file (Git)

Puoi aggiungere file in un repository locale e inviare le modifiche a un CodeCommit repository. Per ulteriori informazioni, consulta [Guida introduttiva a Git e AWS CodeCommit](#).

Modifica il contenuto di un file in un AWS CodeCommit repository

Puoi usare la CodeCommit console o un client Git per modificare il contenuto di un file in un CodeCommit repository. AWS CLI



Argomenti

- [Modifica di un file \(console\)](#)
- [Modifica o elimina un file \(\)AWS CLI](#)
- [Modifica un file \(Git\)](#)

Modifica di un file (console)

È possibile utilizzare la CodeCommit console per modificare un file che è stato aggiunto a un ramo in un CodeCommit repository. Durante la modifica del file, puoi fornire il tuo nome utente e un indirizzo e-mail. Puoi anche aggiungere un messaggio di commit in modo che gli altri utenti sappiano chi ha apportato la modifica e perché.

Per modificare un file in un repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository) selezionare il repository in cui si desidera modificare un file.

3. Nella visualizzazione Code (Codice) selezionare il ramo in cui si desidera modificare il file. Per impostazione predefinita, quando si apre la visualizzazione Code (Codice) vengono visualizzati i contenuti del ramo predefinito.

Per modificare la visualizzazione su un altro ramo, scegliere il pulsante di selezione della visualizzazione. Scegliere il nome di un ramo dall'elenco a discesa oppure immettere il nome del ramo nella casella di filtro e quindi sceglierlo dall'elenco.

4. Esplorare il contenuto del ramo e scegliere il file da modificare. Nella vista file scegliere Edit (Modifica).

Note

Se si sceglie un file binario, viene visualizzato un messaggio di avviso che chiede di confermare che si desidera visualizzare il contenuto. Non dovresti usare la CodeCommit console per modificare file binari.

5. Modificare il file e fornire agli altri utenti informazioni relative a chi ha apportato la modifica e perché.
 - In Author name (Nome creatore), immettere il proprio nome. Questo nome viene utilizzato sia come nome dell'autore che come nome del committer nelle informazioni di commit. CodeCommit utilizza per impostazione predefinita il nome utente IAM o una derivazione dell'accesso alla console come nome dell'autore.
 - In Indirizzo e-mail, inserisci un indirizzo e-mail in modo che altri utenti del repository possano contattarti in merito a questa modifica.
 - In Commit message (Messaggio di commit), immettere una breve descrizione delle modifiche.
6. Scegliere Commit changes (Convalida modifiche) per salvare le modifiche apportate al file e applicare le modifiche al repository.

Modifica o elimina un file ()AWS CLI

È possibile utilizzare il AWS CLI put-file comando and per apportare modifiche a un file in un CodeCommit repository. Puoi anche utilizzare il comando put-file per aggiungere una struttura di directory o un percorso per il file modificato, se desideri archiviare il file modificato in un percorso diverso da quello originale. Se desideri eliminare un file completamente, puoi utilizzare il comando delete-file.

 Note

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per modificare un file in un repository

1. Utilizzando una copia locale del file, apportare le modifiche desiderate da aggiungere al repository CodeCommit.
2. Dal terminale o dalla riga di comando, eseguire il comando `put-file`, specificando:
 - Il repository in cui si desidera aggiungere il file modificato.
 - Il ramo in cui si desidera aggiungere il file modificato.
 - L'ID commit completo dell'ultimo commit effettuato al ramo, noto anche come commit di estremità o testa.
 - Il percorso locale del file.
 - Il nome del file aggiornato che si desidera aggiungere, inclusi il percorso in cui il file è archiviato nel repository, se presente.
 - Il nome utente e l'indirizzo e-mail che si desidera associare a questa modifica del file.
 - Un messaggio di commit che illustra le modifiche apportate.

Il nome utente, l'indirizzo e-mail e il messaggio di commit sono facoltativi, ma consentono agli altri utenti di sapere chi è l'autore e il motivo della modifica. Se non fornisci un nome utente, per CodeCommit impostazione predefinita utilizza il tuo nome utente IAM o una derivazione dell'accesso alla console.

Ad esempio, per aggiungere modifiche apportate a un file denominato a un repository denominato *ExampleSolution.py* a un ramo denominato *MyDemoRepo* il *feature-randomizationfeature* cui commit più recente ha un ID di: *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

Per aggiungere un file binario modificato, è possibile utilizzare `--file-content` con la notazione **fileb**:`//MyDirectory/MyFile.raw`.

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Per eliminare un file, utilizza il comando `delete-file`. Ad esempio, per eliminare un file denominato **README.md** in un ramo denominato **main** con un ID di commit più recente di **c5709475EXAMPLE** in un repository denominato: **MyDemoRepo**

```
aws codecommit delete-file --repository-name MyDemoRepo --branch-name main --file-
path README.md --parent-commit-id c5709475EXAMPLE
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "blobId": "559b44fEXAMPLE",
  "commitId": "353cf655EXAMPLE",
  "filePath": "README.md",
  "treeId": "6bc824cEXAMPLE"
}
```

Modifica un file (Git)

È possibile modificare i file in un repository locale e inviare le modifiche a un CodeCommit repository. Per ulteriori informazioni, consulta [Guida introduttiva a Git e AWS CodeCommit](#).

Lavorare con le richieste pull nei AWS CodeCommit repository

Una richiesta di pull rappresenta per te e gli altri utenti del repository lo strumento principale tramite cui rivedere, commentare e unire le modifiche al codice da un ramo a un altro. È possibile utilizzare le richieste pull per rivedere in modo collaborativo le modifiche minori o le correzioni apportate al codice, le principali funzionalità aggiunte o le nuove versioni del software rilasciato. Questo è un possibile flusso di lavoro per una richiesta di pull:

Li Juan, uno sviluppatore che lavora in un repository denominato MyDemoRepo, desidera lavorare su una nuova funzionalità per una versione imminente di un prodotto. Per mantenere il suo lavoro separato dal codice pronto per la produzione, crea un ramo diverso dal ramo predefinito e gli dà un nome. *feature-randomizationfeature* Scrive codice, esegue commit e infine esegue il push del codice della nuova funzionalità in questo ramo. Desidera che altri utenti del repository rivedano il codice per garantirne la qualità prima dell'unione delle modifiche nel ramo predefinito. A tale scopo, crea una richiesta di pull. Tale richiesta contiene il confronto tra il ramo su cui lavora e quello del codice in cui intende unire le modifiche (in questo caso, il ramo predefinito). È inoltre possibile creare una regola di approvazione che richiede un numero specificato di utenti per approvare la richiesta pull. Può anche specificare un pool di approvazioni di utenti. Altri utenti rivedono il codice e le modifiche da lei apportate, aggiungendo commenti e suggerimenti. È possibile che Li Juan aggiorni più volte il ramo su cui lavora modificando il codice in base ai commenti. Le sue modifiche vengono incorporate nella pull request ogni volta che le invia a quella filiale. CodeCommit Potrebbe anche integrare le modifiche apportate nel ramo di destinazione previsto mentre la richiesta di pull è aperta, in modo da garantire che gli utenti revisionino tutte le modifiche proposte nel contesto. Quando lei e i suoi revisori sono soddisfatti e le condizioni per le regole di approvazione (se presenti) sono state soddisfatte, lei o uno dei suoi revisori unisce il suo codice e chiude la richiesta pull.

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

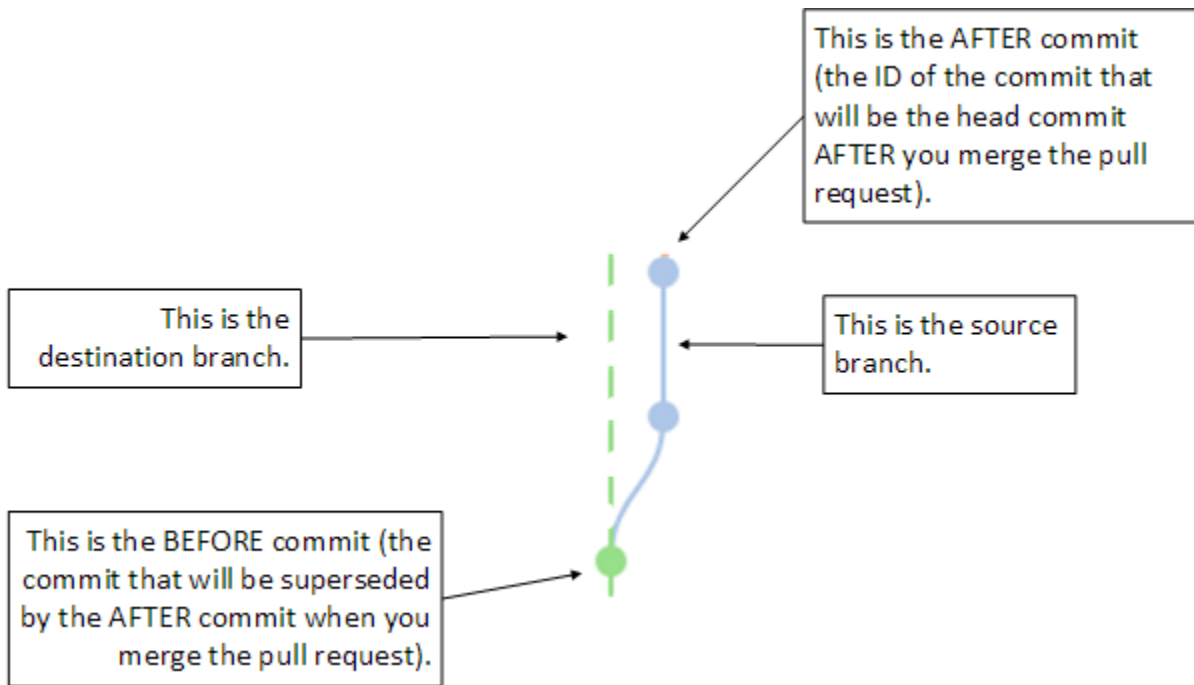
Title
Review changes for bugfix-1236
150 characters maximum

Description - optional Preview markdown [Learn more](#)

I've added some code for the bucket creation issue. Please review by Tuesday.

Changes Commits

Per le richieste pull sono necessari due rami: un ramo di origine che contiene il codice che deve essere rivisto e un ramo di destinazione, dove verrà unito il codice revisionato. Il ramo di origine contiene il commit AFTER, ovvero il commit con le modifiche da unire nel ramo di destinazione. Il ramo di destinazione contiene il commit BEFORE, che rappresenta lo stato del codice prima che il ramo della richiesta di pull venisse unito nel ramo di destinazione. La scelta della strategia di fusione influisce sui dettagli del modo in cui i commit vengono uniti tra i rami di origine e di destinazione nella console. CodeCommit Per ulteriori informazioni sulle strategie di unione in, vedere. CodeCommit [Unisce una pull request \(console\)](#)



La richiesta di pull mostra le differenze tra l'inizio del ramo di origine e l'ultimo commit nel ramo di destinazione quando la richiesta di pull viene creata, in modo da consentire agli utenti di visualizzare e commentare le modifiche. È possibile aggiornare la richiesta di pull in base ai commenti eseguendo il commit e il push delle modifiche al ramo di origine.

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))
  + 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

Quando il codice è stato esaminato e i requisiti della regola di approvazione (se presenti) sono stati soddisfatti, è possibile chiudere la richiesta pull in uno dei diversi modi seguenti:

- Unire i rami in locale ed eseguire il push delle modifiche. Ciò chiude automaticamente la richiesta se viene utilizzata la strategia di unione rapida e non vi sono conflitti di unione.
- Usa la AWS CodeCommit console per chiudere la pull request senza unirla, risolvere i conflitti in caso di unione o, se non ci sono conflitti, chiudere e unire i rami utilizzando una delle strategie di unione disponibili.
- Utilizzo della AWS CLI.

Prima di creare una richiesta di pull:

- Assicurati di aver eseguito il commit e il push a un ramo (ramo di origine) delle modifiche apportate al codice che desideri vengano revisionate.
- Configura le notifiche per il repository, in modo che altri utenti possano essere informati sulla richiesta di pull e sulle modifiche apportate. Questa fase è facoltativa, ma è consigliata.
- Creare e associare i modelli di regola di approvazione al repository, in modo che le regole di approvazione vengano create automaticamente per le richieste pull per garantire la qualità del codice. Per ulteriori informazioni, consulta [Utilizzo dei modelli di regola di approvazione](#).

Le pull request sono più efficaci quando hai configurato gli utenti IAM per gli utenti del tuo repository nel tuo account Amazon Web Services. È più facile identificare quale utente ha fatto un determinato commento. L'altro vantaggio è che gli utenti IAM possono utilizzare le credenziali Git per l'accesso al repository. Per ulteriori informazioni, consulta [Fase 1: Configurazione iniziale per CodeCommit](#). Le richieste pull possono essere utilizzate con altri tipi di utenti, inclusi quelli con accesso federato.

Per informazioni sull'utilizzo di altri aspetti del repository in CodeCommit, consulta, [Lavorare con i repository](#), [Utilizzo dei modelli di regola di approvazione](#), [Lavorare con i file](#), [Lavorare con i commit](#), [Gestione dei rami](#) e [Utilizzo delle preferenze dell'utente](#)

Argomenti

- [Creazione di una richiesta di pull](#)
- [Creare una regola di approvazione per una pull request](#)
- [Visualizza le richieste pull in un AWS CodeCommit repository](#)
- [Revisione di una richiesta di pull](#)
- [Aggiornamento di una richiesta di pull](#)
- [Modifica o elimina una regola di approvazione per una pull request](#)

- [Sostituisci le regole di approvazione su una pull request](#)
- [Unisci una richiesta pull in un repository AWS CodeCommit](#)
- [Risolvi i conflitti in una pull request in un AWS CodeCommit repository](#)
- [Chiudere una pull request in un repository AWS CodeCommit](#)

Creazione di una richiesta di pull

La creazione di richieste pull aiuta altri utenti a vedere ed esaminare le modifiche del codice prima di unirle in un altro ramo. In primo luogo, è necessario creare un ramo per le modifiche del codice che viene definito come il ramo di origine della richiesta pull. Dopo aver eseguito il commit e il push delle modifiche al repository, è possibile creare una richiesta pull che confronta il contenuto di quel ramo (ramo di origine) con il ramo in cui si desidera unire le modifiche dopo la chiusura della richiesta pull (ramo di destinazione).

È possibile utilizzare la AWS CodeCommit console o AWS CLI creare richieste pull per il repository.

Argomenti

- [Crea una pull request \(console\)](#)
- [Crea una pull request \(\)AWS CLI](#)

Crea una pull request (console)

È possibile utilizzare la CodeCommit console per creare una richiesta pull in un CodeCommit repository. Se il repository viene [configurato con le notifiche](#), gli utenti iscritti riceveranno un'e-mail quando crei una richiesta di pull.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera creare una richiesta pull.
3. Nel riquadro di navigazione, selezionare Pull Requests (Richieste pull).

Tip

È anche possibile creare richieste pull da Branches (Rami) e Code (Codice).

4. Scegliere Create pull request (Crea richiesta pull).

Pull requests

Info

Q

Open pull requests

Create pull request

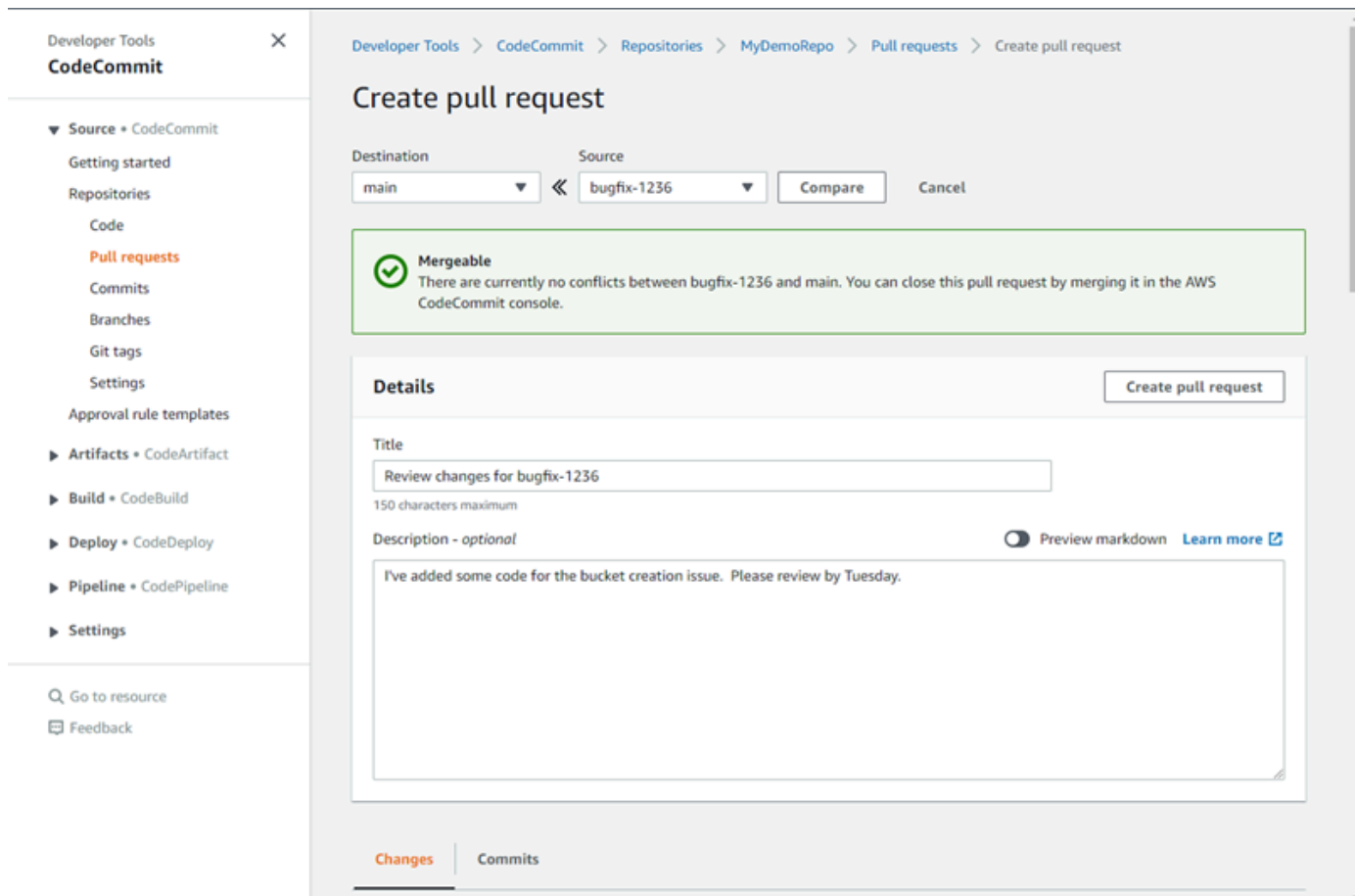
<

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

- In Create pull request (Crea richiesta di pull), in Source (Origine), scegliere il ramo che contiene le modifiche da far esaminare.
- In Destinazione, scegli il ramo in cui intendi unire le modifiche al codice quando la pull request viene chiusa.
- Scegliere Compare (Confronta). Viene eseguito un confronto sui due rami e vengono visualizzate le differenze. Viene inoltre eseguita un'analisi per determinare se i due rami possono essere uniti automaticamente quando viene chiusa la richiesta pull.
- Esaminare i dettagli del confronto e le modifiche per essere sicuri che la richiesta pull contenga le modifiche e i commit desiderati. In caso contrario, modificare le opzioni per i rami di origine e di destinazione e scegliere di nuovo Compare (Confronta).
- Quando i risultati del confronto per la richiesta pull sono soddisfacenti, immettere in Title (Titolo) un titolo breve ma descrittivo per questa revisione. Questo è il titolo che viene visualizzato nell'elenco di richieste pull per il repository.
- (Facoltativo) In Description (Descrizione), immettere i dettagli su questa revisione ed eventuali altre informazioni utili per i revisori.
- Scegli Create (Crea).



La richiesta di pull viene visualizzata nell'elenco di richieste pull per il repository. Se hai [configurato le notifiche](#), gli abbonati all'argomento Amazon SNS ricevono un'e-mail per informarli della richiesta pull appena creata.

Crea una pull request ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Da utilizzare AWS CLI per creare una richiesta pull in un CodeCommit repository

- Eseguire il comando `create-pull-request` specificando:
 - Il nome della richiesta pull (con l'opzione `--title`).
 - La descrizione della richiesta pull (con l'opzione `--description`).
 - Un elenco di destinazioni per il comando `create-pull-request`, tra cui:

- Il nome del CodeCommit repository in cui viene creata la pull request (con l'attributo `repositoryName`).
- Il nome del ramo che contiene le modifiche del codice che si desidera esaminare, noto anche come ramo di origine (con l'attributo `sourceReference`).
- (Facoltativo) Il nome del ramo in cui si intende unire le modifiche al codice, noto anche come il ramo di destinazione, se non si desidera unire al ramo predefinito (con l'attributo `destinationReference`).
- Un token di idempotenza univoco generato dal client (con l'opzione `--client-request-token`).

Questo esempio crea una pull request denominata *Pronunciation difficulty analyzer* con una descrizione *Please review these changes by Tuesday* che si rivolge al ramo di *jane-branch* origine. La pull request deve essere unita al ramo predefinito *main* in un CodeCommit repository denominato: *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
```

Creare una regola di approvazione per una pull request

La creazione di regole di approvazione per le richieste pull consente di garantire la qualità del codice richiedendo agli utenti di approvare la richiesta pull prima che il codice possa essere unito nel ramo di destinazione. È possibile specificare il numero di utenti che devono approvare una richiesta pull. È inoltre possibile specificare un pool di approvazioni di utenti per la regola. In tal caso, solo le approvazioni di tali utenti vengono conteggiate per il numero di approvazioni richieste per la regola.

Note

Puoi anche creare modelli di regole di approvazione, che possono aiutarti ad automatizzare la creazione di regole di approvazione tra i repository che verranno applicate a tutte le pull request. Per ulteriori informazioni, consulta [Utilizzo dei modelli di regola di approvazione](#).

Puoi utilizzare la AWS CodeCommit console o il AWS CLI per creare regole di approvazione per il tuo repository.

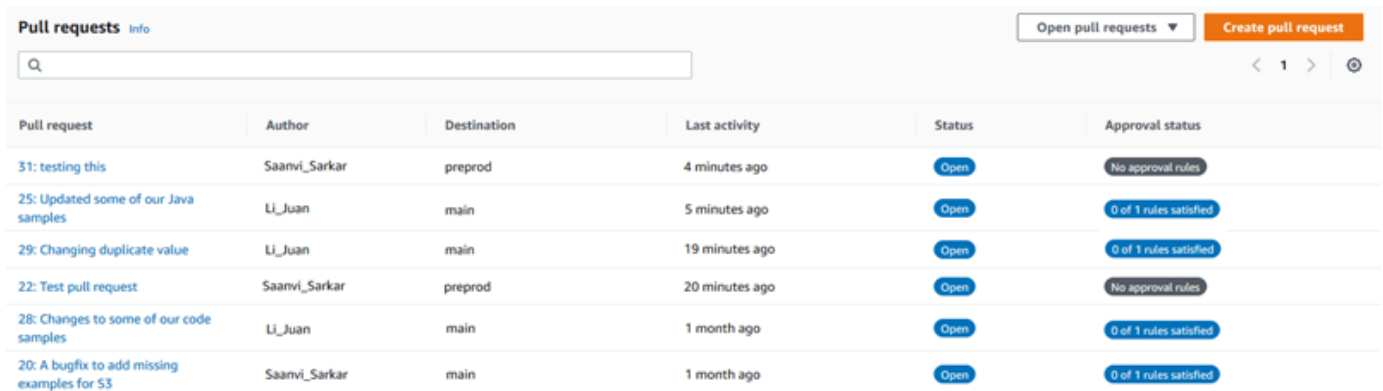
Argomenti

- [Crea una regola di approvazione per una pull request \(console\)](#)
- [Crea una regola di approvazione per una pull request \(\)AWS CLI](#)

Crea una regola di approvazione per una pull request (console)

È possibile utilizzare la CodeCommit console per creare una regola di approvazione per una pull request in un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera creare una richiesta pull.
3. Nel riquadro di navigazione, selezionare Pull Requests (Richieste pull).
4. Scegliere la richiesta pull per la quale si desidera creare una regola di approvazione dall'elenco. È possibile creare regole di approvazione solo per le richieste pull aperte.



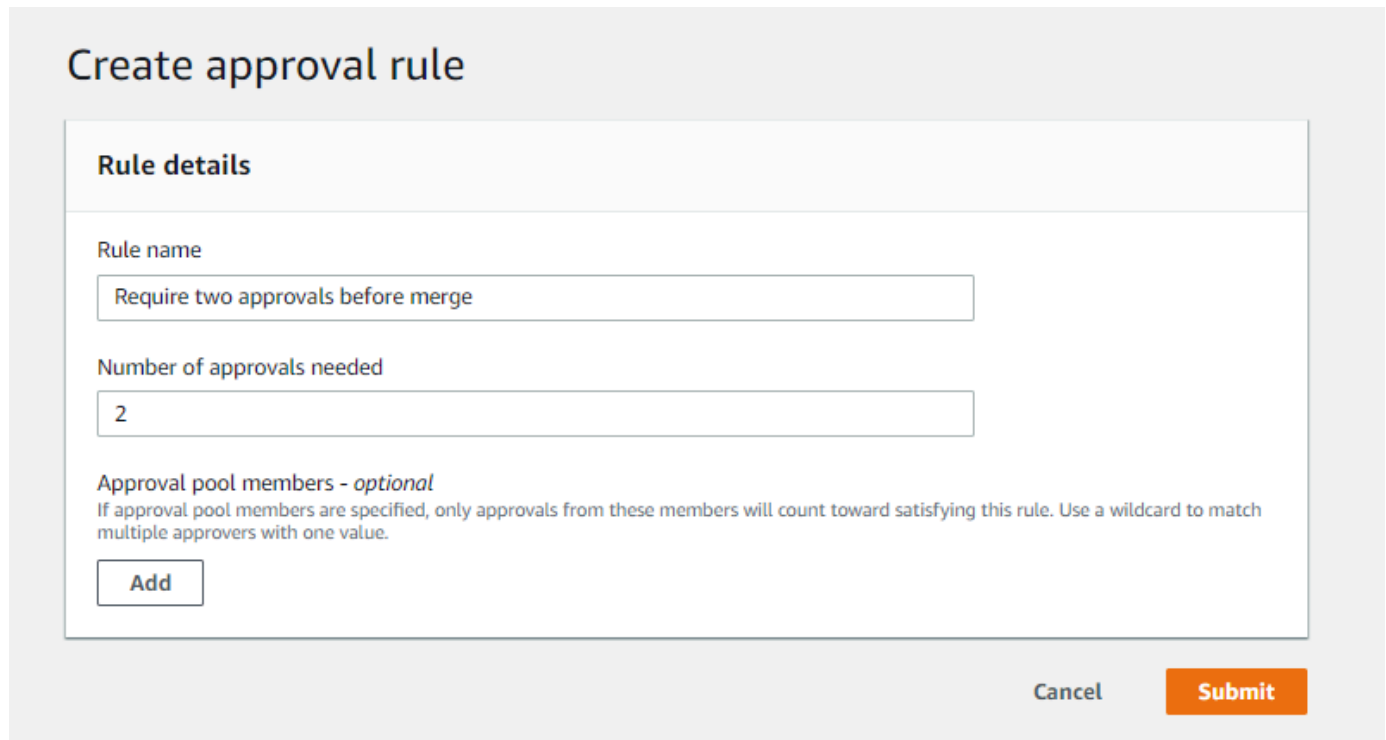
Pull request	Author	Destination	Last activity	Status	Approval status
31: testing this	Saanvi_Sarkar	preprod	4 minutes ago	Open	No approval rules
25: Updated some of our Java samples	Li_Juan	main	5 minutes ago	Open	0 of 1 rules satisfied
29: Changing duplicate value	Li_Juan	main	19 minutes ago	Open	0 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. Nella richiesta pull, scegliere Approvals (Approvazioni), quindi scegliere Create approval rule (Crea regola di approvazione).
6. In Rule name (Nome regola), assegnare alla regola un nome descrittivo in modo da sapere a cosa serve. Ad esempio, se si desidera che due persone approvino una richiesta pull prima che possa essere unita, è possibile assegnare un nome alla regola **Require two approvals before merge**.

Note

Non è possibile modificare il nome di una regola di approvazione dopo averla creata.

In Number of approvals needed (Numero di approvazioni necessarie), immettere il numero desiderato. Il valore di default è 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. (Facoltativo) Se si desidera richiedere che le approvazioni per una richiesta pull provengano da un gruppo specifico di utenti, in Approval rule members (Membri della regola di approvazione), scegliere Add (Aggiungi). In Approver type (Tipo approvatore), scegliere una delle seguenti opzioni:

- Nome utente IAM o ruolo assunto: questa opzione precompila l'ID dell' AWS account con l'account che hai usato per accedere e richiede solo un nome. Può essere utilizzato sia per gli utenti IAM che per gli utenti ad accesso federato il cui nome corrisponde al nome fornito. Questa è un'opzione molto potente che offre una grande flessibilità. Ad esempio, se hai effettuato l'accesso con l'account Amazon Web Services 123456789012 e scegli questa opzione e specifichi **Mary_Major**, tutte le seguenti approvazioni vengono conteggiate come approvazioni provenienti da quell'utente:
 - Un utente IAM nell'account () `arn:aws:iam::123456789012:user/Mary_Major`
 - Un utente federato identificato in IAM come Mary_Major ()
`arn:aws:sts::123456789012:federated-user/Mary_Major`

Questa opzione non riconoscerebbe una sessione attiva di qualcuno che assume il ruolo di **CodeCommitReview** con un nome di sessione di ruolo `Mary_Major` (`arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major`) a meno che non si includa un carattere jolly (`*Mary_Major`). È inoltre possibile specificare esplicitamente il nome del ruolo (`CodeCommitReview/Mary_Major`).

- **ARN completo:** questa opzione consente di specificare l'Amazon Resource Name (ARN) completo dell'utente o del ruolo IAM. Questa opzione supporta anche i ruoli presunti utilizzati da altri AWS servizi, come e AWS Lambda . AWS CodeBuild Per i ruoli assunti, il formato ARN deve essere `arn:aws:sts::AccountID:assumed-role/RoleName` per i ruoli e `arn:aws:sts::AccountID:assumed-role/FunctionName` per le funzioni.

Se hai scelto il nome utente IAM o hai assunto il ruolo come tipo di approvatore, in Valore inserisci il nome dell'utente o del ruolo IAM o l'ARN completo dell'utente o del ruolo. Scegliere Add (Aggiungi) di nuovo per aggiungere altri utenti o ruoli, fino a quando non saranno stati aggiunti tutti gli utenti o i ruoli le cui approvazioni valgono per il numero di approvazioni richieste.

Entrambi i tipi di approvatore consentono di utilizzare caratteri jolly (*) nei loro valori. Ad esempio, se scegli l'opzione nome utente IAM o ruolo assunto e lo specifichi **CodeCommitReview/***, tutti gli utenti che assumono il ruolo di **CodeCommitReview** vengono conteggiati nel pool di approvazione. I nomi delle sessioni di ruolo individuali valgono per il numero richiesto di approvatori. In questo modo, sia `Mary_Major` che `Li_Juan` sono contati come approvazioni al momento dell'accesso e dell'assunzione del ruolo di `CodeCommitReview`. Per ulteriori informazioni su IAM ARNs, wildcard e formati, consulta [IAM Identifiers](#).

Note

Le regole di approvazione non supportano le approvazioni cross-account.

8. Al termine della configurazione della regola di approvazione, scegliere Submit (Invia).

Crea una regola di approvazione per una pull request ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per creare una regola di approvazione per una pull request in un CodeCommit repository

- Eseguire il comando `create-pull-request-approval-rule` specificando:
 - L'ID della richiesta di pull (con l'opzione `--id`).
 - Il nome della regola di approvazione (con l'opzione `--approval-rule-name`).
 - Il contenuto della regola di approvazione (con l'opzione `--approval-rule-content`).

Quando si crea la regola di approvazione, è possibile specificare gli approvatori in un pool di approvazioni in due modi:

- **CodeCommitApprovers**: questa opzione richiede solo un account Amazon Web Services e una risorsa. Può essere utilizzata sia per gli utenti IAM che per gli utenti ad accesso federato il cui nome corrisponde al nome della risorsa fornito. Questa è un'opzione molto potente che offre una grande flessibilità. Ad esempio, se specifichi l'account Amazon Web Services 123456789012 e **Mary_Major** tutte le seguenti approvazioni vengono conteggiate come approvazioni provenienti da quell'utente:
 - Un utente IAM nell'account () `arn:aws:iam::123456789012:user/Mary_Major`
 - Un utente federato identificato in IAM come `Mary_Major` ()
`arn:aws:sts::123456789012:federated-user/Mary_Major`

Questa opzione non riconoscerebbe una sessione attiva di qualcuno che assume il ruolo di **CodeCommitReview** con un nome di sessione di ruolo `Mary_Major` (`arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major`) a meno che non si includa un carattere jolly (`*Mary_Major`).

- **ARN completo**: questa opzione consente di specificare l'Amazon Resource Name (ARN) completo dell'utente o del ruolo IAM.

[Per ulteriori informazioni su IAM ARNs, i wildcard e i formati, consulta IAM Identifiers.](#)

Nell'esempio seguente viene creata una regola di approvazione denominata `Require two approved approvers` per una richiesta pull con l'ID di 27. La regola specifica che sono necessarie due approvazioni da un pool di approvazioni. Il pool include tutti gli utenti che accedono CodeCommit e assumono il ruolo di **CodeCommitReview** nell'account 123456789012 Amazon Web Services. Include anche un utente IAM o un utente federato denominato `Nikhil_Jayashankar` nello stesso account 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/*\"]}]}"
```

Visualizza le richieste pull in un AWS CodeCommit repository

È possibile utilizzare la AWS CodeCommit console o visualizzare le AWS CLI richieste pull per il proprio repository. Per impostazione predefinita, è possibile visualizzare solo le richieste pull aperte, ma è possibile modificare il filtro per visualizzare tutte le richieste pull, solo le richieste chiuse, solo le richieste pull create da te e molto altro.

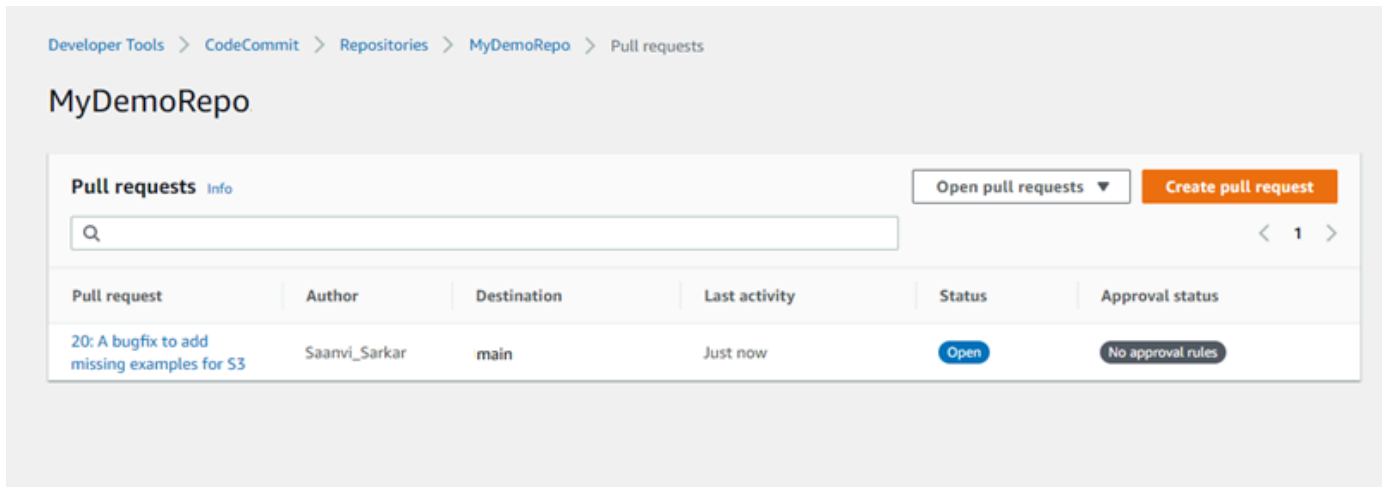
Argomenti

- [Visualizza le richieste pull \(console\)](#)
- [Visualizza le richieste pull \(AWS CLI\)](#)

Visualizza le richieste pull (console)

È possibile utilizzare la AWS CodeCommit console per visualizzare un elenco di richieste pull in un CodeCommit repository. Modificando il filtro, puoi modificare la visualizzazione dell'elenco per vedere solo determinati set di richieste pull. Ad esempio, puoi visualizzare una lista di richieste pull create con uno stato Open (Aperto) oppure puoi scegliere un altro filtro e visualizzare le richieste pull create con uno stato Closed (Chiuso).

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera visualizzare le richieste pull.
3. Nel riquadro di navigazione, selezionare Pull Requests (Richieste pull).
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte.



5. Per modificare il filtro di visualizzazione, sceglierlo dall'elenco dei filtri disponibili:
 - Open pull requests (Apri richieste aperte) (impostazione predefinita): visualizza tutte le richieste pull con stato Open (Aperto).
 - All pull requests (Tutte le richieste pull): visualizza tutte le richieste pull.
 - Closed pull requests (Richieste pull chiuse): visualizza tutte le richieste pull con stato Closed (Chiuso).
 - My pull requests (Le mie richieste di pull): consente di visualizzare tutte le richieste di pull create, indipendentemente dallo stato. Non visualizza le recensioni che hai commentato o a cui hai partecipato.
 - My open pull requests (Le mie richieste pull aperte): visualizza tutte le richieste pull create con uno stato Open (Aperto).
 - My closed pull requests (Le mie richieste pull chiuse): visualizza tutte le richieste pull create con uno stato Closed (Chiuso).
6. Quando si trova una richiesta pull nell'elenco visualizzato che si desidera visualizzare, selezionarla.

Visualizza le richieste pull (AWS CLI)

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Segui questi passaggi per utilizzare AWS CLI per visualizzare le richieste pull in un CodeCommit repository.

1. Per visualizzare un elenco di richieste pull in un repository, eseguire il comando `list-pull-requests`, specificando:
 - Il nome del CodeCommit repository in cui desideri visualizzare le richieste pull (con l'opzione `--repository-name`).
 - (Facoltativo) Lo stato della richiesta pull (con l'opzione `--pull-request-status`).
 - (Facoltativo) L'Amazon Resource Name (ARN) dell'utente IAM che ha creato la pull request (con l'opzione `--author-arn`).
 - (Facoltativo) Un token di enumerazione può essere utilizzato per restituire il batch dei risultati (con l'opzione `--next-token`).
 - (Facoltativo) Una restrizione sul numero di risultati restituiti per richiesta (con l'opzione `--max-results`).

Ad esempio, per elencare le richieste pull create da un utente IAM con l'ARN `arn:aws:iam::111111111111:user/Li_Juan` e lo stato di `CLOSED` in un CodeCommit repository denominato: `MyDemoRepo`

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "nextToken": "",
  "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

Le pull request IDs vengono visualizzate nell'ordine delle attività più recenti.

2. Per visualizzare i dettagli di una richiesta pull, eseguire il comando `get-pull-request` con l'opzione `--pull-request-id`, specificando l'ID della richiesta pull. Ad esempio, per visualizzare informazioni su una pull request con l'ID di `27`:

```
aws codecommit get-pull-request --pull-request-id 27
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.

Per visualizzare le approvazioni su una richiesta pull, eseguire il comando `get-pull-request-approval-state`, specificando:

- L'ID della richiesta di pull (da utilizzare con l'opzione `--pull-request-id`).
- L'ID di revisione della richiesta pull, utilizzando il `--revision-id` option). È possibile ottenere l'ID di revisione corrente per una richiesta pull utilizzando il comando [get-pull-request](#).

Ad esempio, per visualizzare le approvazioni di una pull request con un ID di **8** e un ID di revisione pari a: **9f29d167EXAMPLE**

```
aws codecommit get-pull-request-approval-state --pull-request-id 8 --revision-id 9f29d167EXAMPLE
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "approvals": [
    {
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
      "approvalState": "APPROVE"
    }
  ]
}
```

4. Per visualizzare gli eventi in una richiesta pull, eseguire il comando `describe-pull-request-events` con l'opzione `--pull-request-id`, specificando l'ID della richiesta pull. Ad esempio, per visualizzare gli eventi di una pull request con l'ID di: **8**

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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. Per vedere eventuali conflitti uniti per una richiesta pull, eseguire il comando `get-merge-conflicts`, specificando:

- Il nome del CodeCommit repository (con l'opzione `--repository-name`).
- Il ramo, il tag, l'HEAD o qualunque altro tipo di riferimento completo per l'origine delle modifiche da utilizzare nella valutazione dell'unione (con l'opzione `--source-commit-specifier`).
- Il ramo, il tag, l'HEAD o qualunque altro tipo di riferimento completo per la destinazione delle modifiche da utilizzare nella valutazione dell'unione (con l'opzione `--destination-commit-specifier`).
- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).

Ad esempio, per vedere se ci sono conflitti di fusione tra la punta di un ramo di origine denominato *my-feature-branch* e un ramo di destinazione denominato *main* in un repository denominato: `MyDemoRepo`

```

aws codecommit get-merge-conflicts --repository-name MyDemoRepo --source-commit-specifier my-feature-branch --destination-commit-specifier main --merge-option FAST_FORWARD_MERGE

```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```

{
  "destinationCommitId": "fac04518EXAMPLE",
  "mergeable": false,
  "sourceCommitId": "16d097f03EXAMPLE"
}

```

Revisione di una richiesta di pull

Puoi usare la AWS CodeCommit console per rivedere le modifiche incluse in una pull request. Puoi aggiungere commenti alla richiesta, ai file e a singole righe di codice. È anche possibile rispondere ai commenti degli altri utenti. Se il repository viene [configurato con le notifiche](#), riceverai e-mail quando gli utenti rispondono ai tuoi commenti o quando gli utenti commentano una richiesta pull.

Puoi usare il AWS CLI per commentare una pull request e rispondere ai commenti. Per rivedere le modifiche, è necessario utilizzare la CodeCommit console, il git diff comando o uno strumento diff.

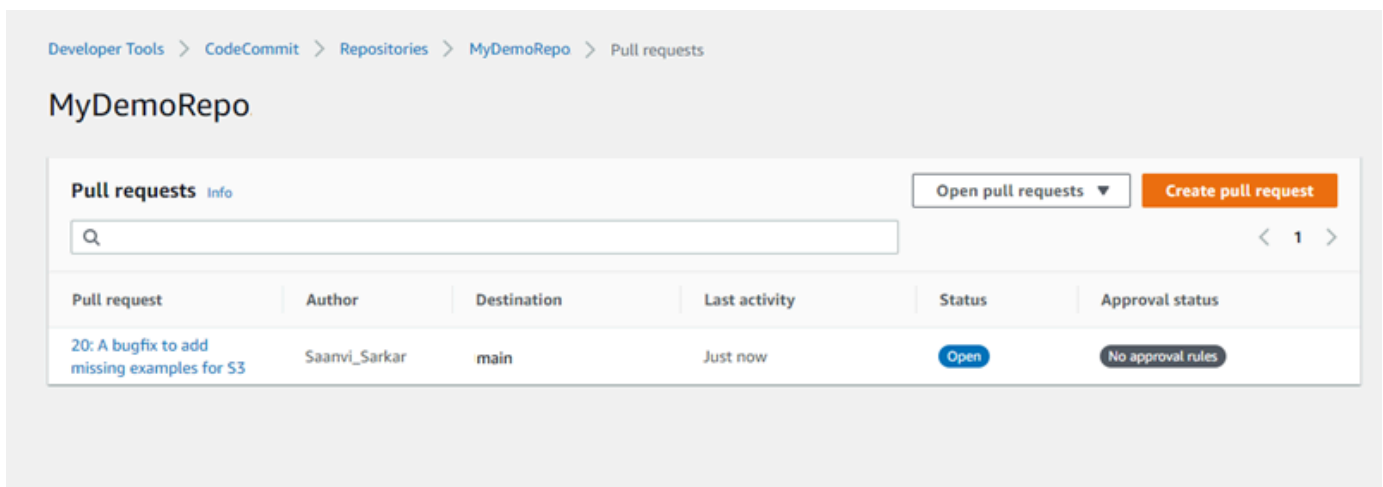
Argomenti

- [Rivedi una pull request \(console\)](#)
- [Rivedi le richieste pull \(AWS CLI\)](#)

Rivedi una pull request (console)

È possibile utilizzare la CodeCommit console per esaminare una richiesta pull in un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), scegliere il nome del repository.
3. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull).
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte. Scegliere la richiesta pull aperta che si desidera rivedere.



 Note

È possibile commentare una richiesta pull chiusa o unita, ma non è possibile unirla o riapirla.

5. Nella richiesta pull, scegliere **Changes (Modifiche)**.
6. Esegui una di queste operazioni:
 - Per aggiungere un commento generale per l'intera richiesta pull, in **Comments on changes** (Commenti sulle modifiche), in **New comment** (Nuovo commento), immettere un commento e selezionare **Save** (Salva). È possibile utilizzare [Markdown](#) o immettere il commento in testo normale.



Comments on changes

New comment Preview markdown [Learn more](#)

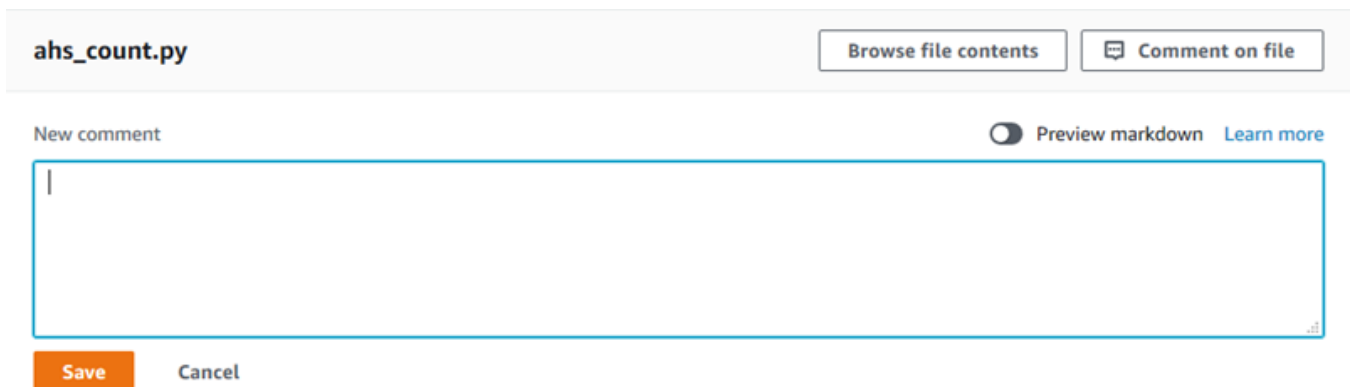
Did we also change the variable name in blf.py and concat.py?

Save

- Per aggiungere un commento a un file nel commit, in **Changes (Modifiche)**, individuare il nome del file. Scegliere l'icona del commento accanto al nome del file



immettere un commento, quindi scegliere **Save** (Salva).



ahs_count.py Browse file contents  Comment on file

New comment Preview markdown [Learn more](#)

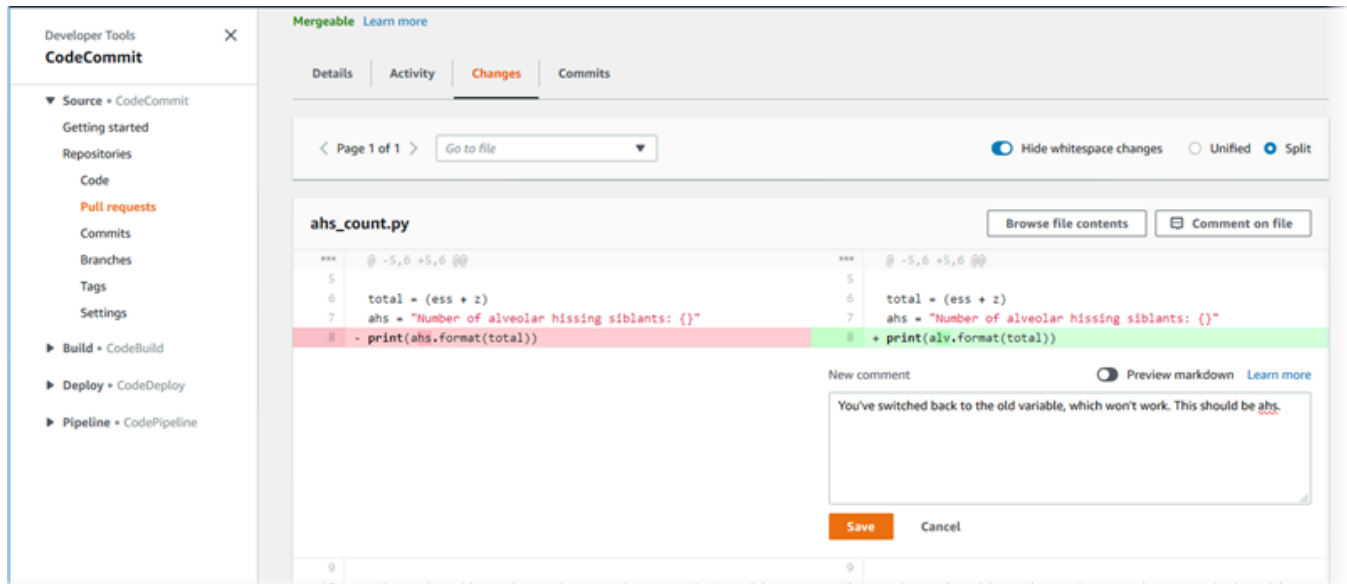
|

Save Cancel

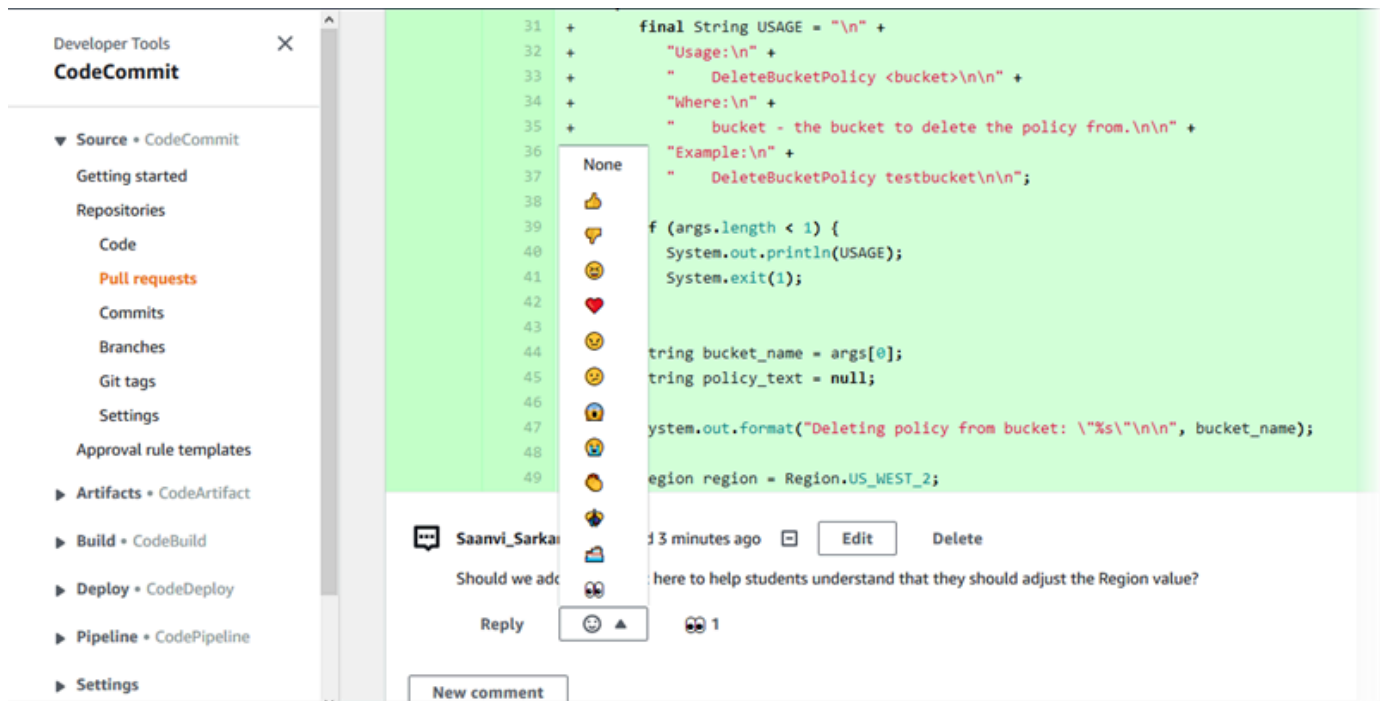
- Per aggiungere un commento a una riga modificata nella richiesta pull, in Changes (Modifiche), vai alla riga che desideri commentare. Scegliere l'icona del commento



visualizzata per quella riga, immettere un commento, quindi scegliere Save (Salva).



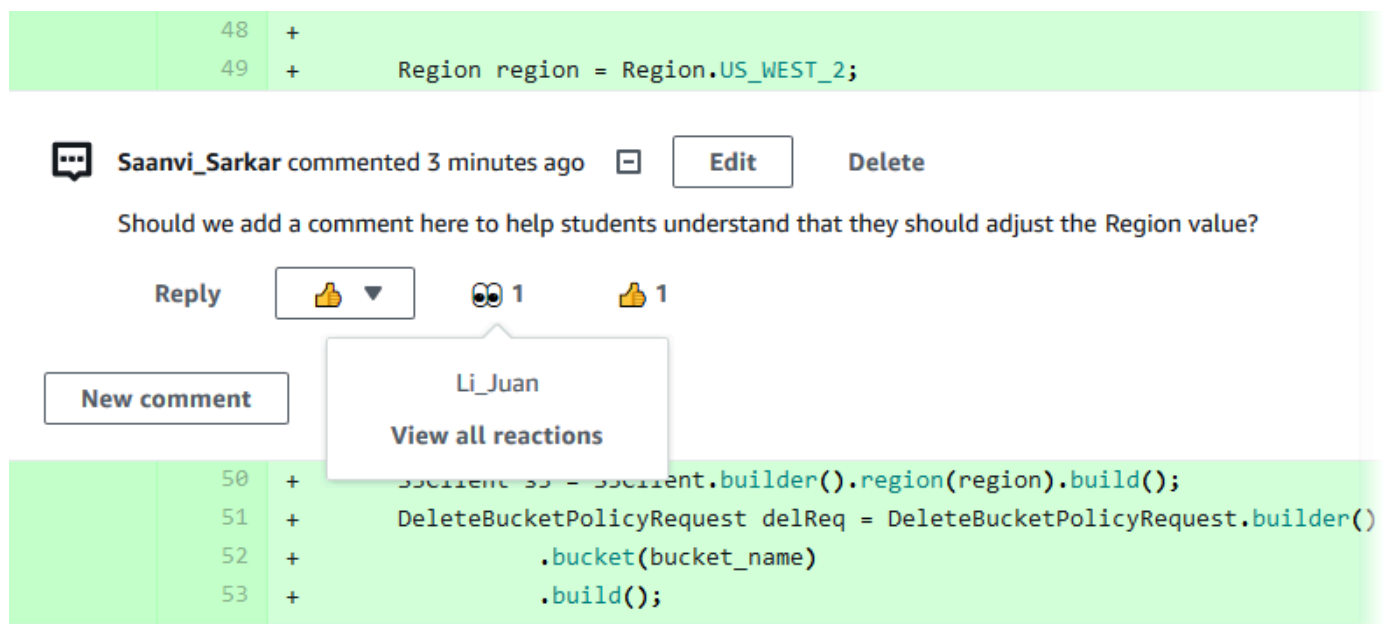
- Per rispondere a commenti su un commit in Changes (Modifiche) o Activity (Attività), scegliere Reply (Rispondi). Puoi rispondere con testo e con emoji.



Puoi visualizzare i nomi di coloro che hanno risposto con una particolare risposta emoji selezionandola. Per visualizzare tutte le reazioni emoji e le informazioni su chi ha risposto con quali emoji, scegli Visualizza tutte le reazioni. Se hai risposto con un'emoji a un commento, la tua risposta viene mostrata nell'icona del pulsante di reazione alle emoji.

Note

I conteggi delle reazioni visualizzati nella console sono accurati al momento del caricamento della pagina. Per le informazioni più aggiornate sul conteggio delle reazioni alle emoji, aggiorna la pagina o scegli Visualizza tutte le reazioni.

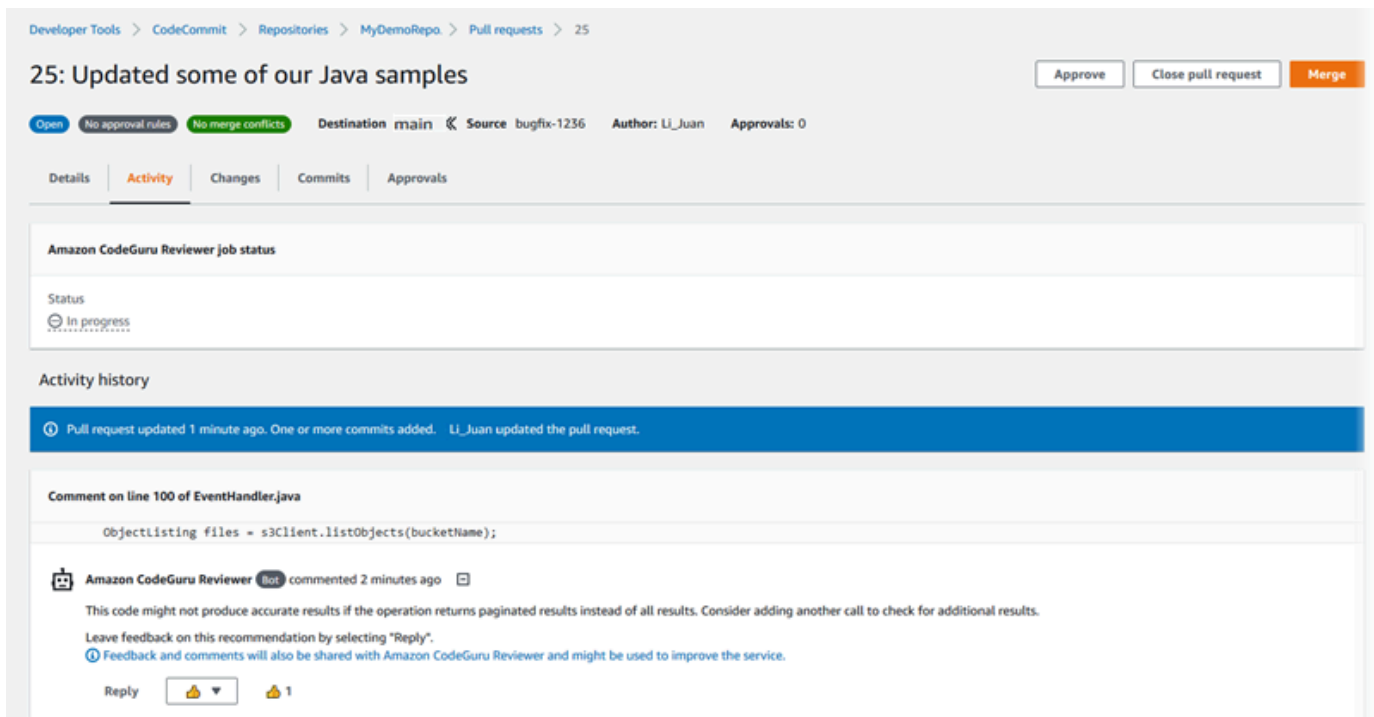


8. (Facoltativo) Per rispondere a un consiglio creato da Amazon CodeGuru Reviewer, incluso fornire un feedback sulla qualità del consiglio, scegli Rispondi. Utilizzare i pulsanti di reazione per fornire informazioni generali sull'approvazione o disapprovazione del suggerimento. Utilizzare il campo commento per fornire maggiori dettagli sulla reazione.

Note

Amazon CodeGuru Reviewer è un servizio automatizzato di revisione del codice che utilizza l'analisi dei programmi e l'apprendimento automatico per rilevare problemi comuni e consigliare correzioni nel codice Java o Python.

- I commenti di Amazon CodeGuru Reviewer vengono visualizzati solo se hai associato il repository ad Amazon CodeGuru Reviewer, se l'analisi è completa e se il codice nella pull request è codice Java o Python. Per ulteriori informazioni, consulta [Associare o dissociare un AWS CodeCommit repository con Amazon Reviewer CodeGuru](#).
- I commenti di Amazon CodeGuru Reviewer vengono visualizzati nella scheda Modifiche solo se sono stati inseriti nella revisione più recente della pull request. Vengono sempre visualizzati nella scheda Attività.
- Sebbene tu possa rispondere con una qualsiasi delle reazioni emoji disponibili ai consigli di Amazon CodeGuru Reviewer, per valutare l'utilità del consiglio vengono utilizzate solo le reazioni emoji pollice su e pollice giù.



9. Per approvare le modifiche apportate in una richiesta pull, scegliere Approve (Approva).

Note

Non è possibile approvare una richiesta pull creata.

È possibile visualizzare le approvazioni, le regole di approvazione per una richiesta pull e le regole di approvazione create dai modelli di regola di approvazione in Approvals (Approvazioni). Se decidi di non approvare la richiesta pull, puoi scegliere Revoke approval (Revoca approvazione).

Note

È possibile approvare o revocare l'approvazione solo su una richiesta di pull aperta. Non è possibile approvare o revocare l'approvazione su una richiesta pull il cui stato è unito o chiuso.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 14

14: Hotfix for customer issue 1234

Revoke approval Close pull request Merge

Open Approved Mergeable Destination master << Source Author: Approvals: 2

Details Activity Changes Commits Approvals

Approvals

Approver	Status
Li_Juan	Approved
Saanvi_Sarkar	Approved

Approval rules

Delete Edit Create approval rule

Approval rule	Status
My Approval Rule	Rule satisfied

Rivedi le richieste pull ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

È possibile esaminare le richieste pull con i seguenti AWS CLI comandi:

- [post-comment-for-pull-request](#), per aggiungere un commento a una pull request
- [get-comments-for-pull-request](#), per visualizzare i commenti lasciati su una pull request

- [update-pull-request-approval-state](#), per approvare o revocare l'approvazione per una pull request
- [post-comment-reply](#), per rispondere a un commento in una pull request

Puoi anche usare emoji con commenti in una pull request con i seguenti comandi:

- Per rispondere a un commento con un'emoji, esegui. [put-comment-reaction](#)
- Per visualizzare le reazioni emoji a un commento, esegui. [get-comment-reactions](#)

Da utilizzare per AWS CLI esaminare le richieste pull in un repository CodeCommit

1. Per aggiungere un commento a una richiesta pull in un repository, eseguire il comando `post-comment-for-pull-request`, specificando:
 - L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
 - Il nome del repository che contiene la richiesta pull (con l'opzione `--repository-name`).
 - L'ID commit completo nel ramo di destinazione a cui viene unita la richiesta pull (con l'opzione `--before-commit-id`).
 - L'ID commit completo del commit nel ramo di origine che è l'estremità corrente del ramo per la richiesta pull quando posti il commento (con l'opzione `--after-commit-id`).
 - Un token di idempotenza univoco generato dal client (con l'opzione `--client-request-token`).
 - Il contenuto del commento (con l'opzione `--content`).
 - Un elenco di informazioni sulla posizione in cui inserire il commento, tra cui:
 - Il nome del file che viene confrontato, inclusa l'estensione e la sottodirectory, se presente (con l'attributo `filePath`).
 - Il numero di riga della modifica in un file confrontato (con l'attributo `filePosition`).
 - Se il commento sulla modifica è "before" o "after" nel confronto tra i rami di origine e di destinazione (con l'attributo `relativeFileVersion`).

Ad esempio, utilizzate questo comando per aggiungere il commento *"These don't appear to be used anywhere. Can we remove them?"* sulla modifica al `ahs_count.py` file in una richiesta pull con l'ID di **47** in un repository denominato *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
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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. Per visualizzare i commenti per una richiesta pull, eseguire il comando `get-comments-for-pull-request` specificando:
 - Il nome del CodeCommit repository (con l'opzione `--repository-name`).
 - L'ID generato dal sistema della richiesta pull (con l'opzione `--pull-request-id`).
 - (Facoltativo) Un token di enumerazione per restituire il batch dei risultati successivo (con l'opzione `--next-token`).
 - (Facoltativo) Un numero intero non negativo per limitare il numero di risultati restituiti (con l'opzione `--max-results`).

Ad esempio, utilizzare questo comando per visualizzare i commenti per una richiesta pull con un ID 42.

```
aws codecommit get-comments-for-pull-request --pull-request-id 42
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.

Per approvare o revocare l'approvazione di una richiesta pull, eseguire il comando `update-pull-request-approval-state`, specificando:

- L'ID della richiesta di pull (da utilizzare con l'opzione `--pull-request-id`).
- L'ID di revisione della richiesta pull, utilizzando il `--revision-id` option). È possibile ottenere l'ID di revisione corrente per una richiesta pull utilizzando il comando [get-pull-request](#).
- Lo stato di approvazione che si desidera applicare, utilizzando l'opzione `--approval-state`. Gli stati di approvazione validi includono APPROVE e REVOKE.

Ad esempio, utilizzate questo comando per approvare una pull request con l'ID di **27** e l'ID di revisione di. **9f29d167EXAMPLE**

```
aws codecommit update-pull-request-approval-state --pull-request-id 27 --revision-id 9f29d167EXAMPLE --approval-state "APPROVE"
```

In caso di successo, questo comando non restituisce alcun risultato.

4. Per pubblicare la risposta a un commento in una richiesta pull, eseguire il comando `post-comment-reply`, specificando:

- L'ID generato dal sistema del commento a cui si desidera rispondere (con l'opzione `--in-reply-to`).
- Un token di idempotenza univoco generato dal client (con l'opzione `--client-request-token`).

- Il contenuto della risposta (con l'opzione `--content`).

Ad esempio, utilizzate questo comando per aggiungere la risposta *"Good catch. I'll remove them."* al commento con l'ID generato dal sistema di. *abcd1234EXAMPLEb5678efgh*

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --
content "Good catch. I'll remove them." --client-request-token 123Example
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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": []
  }
}
```

Aggiornamento di una richiesta di pull

È possibile aggiornare una pull request con ulteriori modifiche al codice inviando i commit al ramo sorgente di una pull request aperta. Per ulteriori informazioni, consulta [Crea un commit in AWS CodeCommit](#).

Puoi usare la AWS CodeCommit console o il AWS CLI per aggiornare il titolo o la descrizione di una pull request. Potresti voler aggiornare il titolo o la descrizione della pull request perché:

- Gli altri utenti non comprendono la descrizione o il titolo originale è fuorviante.
- Se desideri che il titolo o la descrizione riflettano le modifiche apportate al ramo d'origine di una richiesta pull aperta.

Aggiorna una pull request (console)

È possibile utilizzare la CodeCommit console per aggiornare il titolo e la descrizione di una pull request in un CodeCommit repository. Per aggiornare il codice nella pull request, invia i commit al ramo sorgente di una pull request aperta.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera aggiornare una richiesta pull.
3. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull).
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte. Scegliere la richiesta pull aperta che si desidera aggiornare.
5. Nella richiesta pull, scegliere Details (Dettagli), quindi scegliere Edit details (Modifica dettagli) per modificare il titolo o la descrizione.

Note

Non è possibile aggiornare il titolo o la descrizione di una richiesta pull chiusa o unita.

Aggiorna le richieste pull ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Potresti anche essere interessato ai seguenti comandi:

- [update-pull-request-approval-state](#), per approvare o revocare l'approvazione su una richiesta pull.
- [create-pull-request-approval-rule](#), per creare una regola di approvazione per una richiesta pull.
- [delete-pull-request-approval-rule](#), per eliminare una regola di approvazione per una richiesta pull.
- [Crea un commit usando il AWS CLI](#) oppure [Crea un commit usando un client Git](#), per creare e inviare ulteriori modifiche al codice nel ramo sorgente di una pull request aperta.

Da utilizzare AWS CLI per aggiornare le richieste pull in un CodeCommit repository

1. Per aggiornare il titolo di una richiesta pull in un repository, eseguire il comando update-pull-request-title specificando:

- L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
- Il titolo della richiesta pull (con l'opzione `--title`).

Ad esempio, per aggiornare il titolo di una pull request con l'ID di **47**:

```
aws codecommit update-pull-request-title --pull-request-id 47 --title  
"Consolidation of global variables - updated review"
```

2. Per aggiornare la descrizione di una richiesta pull, eseguire il comando `update-pull-request-description` specificando:

- L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
- La descrizione (con l'opzione `--description`).

Ad esempio, per aggiornare la descrizione di una pull request con l'ID di **47**:

```
aws codecommit update-pull-request-description --pull-request-id 47 --description  
"Updated the pull request to remove unused global variable."
```

Modifica o elimina una regola di approvazione per una pull request

Quando si dispone di una regola di approvazione in una richiesta pull, non è possibile unire tale richiesta fino a quando non sono state soddisfatte le condizioni. È possibile modificare le regole di approvazione per le richieste pull per semplificare la soddisfazione delle loro condizioni o per aumentare il rigore delle revisioni. È possibile modificare il numero di utenti che devono approvare una richiesta pull. È inoltre possibile aggiungere, rimuovere o modificare l'appartenenza in un pool di approvazioni di utenti per la regola. Infine, se non si desidera più utilizzare una regola di approvazione per una richiesta pull, è possibile eliminarla.

Note

È inoltre possibile ignorare le regole di approvazione per una richiesta pull. Per ulteriori informazioni, consulta [Sostituisci le regole di approvazione su una pull request](#).

Puoi utilizzare la AWS CodeCommit console o AWS CLI per modificare ed eliminare le regole di approvazione per il tuo repository.

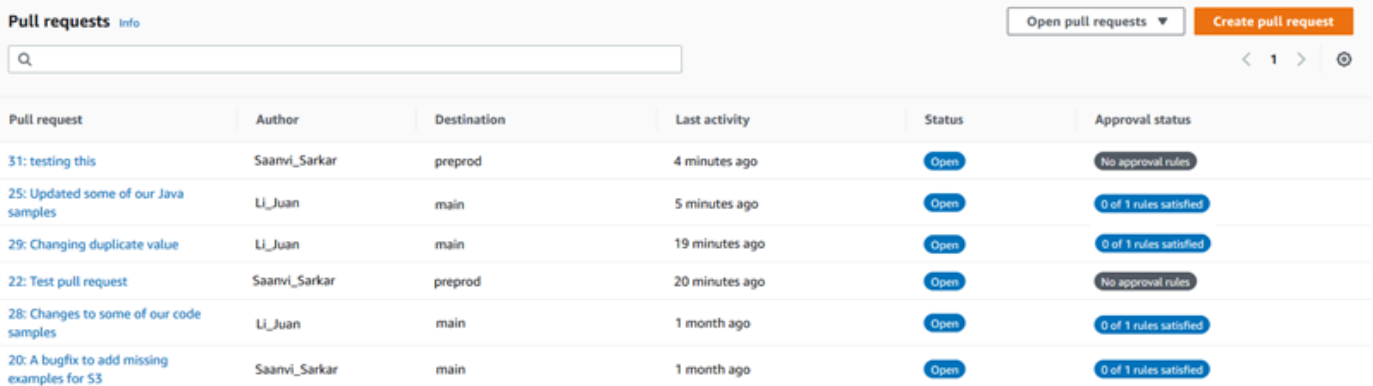
Argomenti

- [Modifica o elimina una regola di approvazione per una pull request \(console\)](#)
- [Modifica o elimina una regola di approvazione per una pull request \(AWS CLI\)](#)

Modifica o elimina una regola di approvazione per una pull request (console)

È possibile utilizzare la CodeCommit console per modificare o eliminare una regola di approvazione per una pull request in un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera modificare o eliminare una regola di approvazione per una richiesta pull.
3. Nel riquadro di navigazione, selezionare Pull Requests (Richieste pull).
4. Scegliere la richiesta pull in cui si desidera modificare o eliminare una regola di approvazione. È possibile modificare ed eliminare solo le regole di approvazione per le richieste pull aperte.



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. Nella richiesta pull, scegliere Approvals (Approvazioni), quindi scegliere la regola che si desidera modificare o eliminare dall'elenco. Esegui una di queste operazioni:
 - Se desideri modificare la regola, scegli Modifica.
 - Se si desidera eliminare la regola, scegliere Delete (Elimina), quindi seguire le istruzioni per verificare l'eliminazione della regola.
6. In Edit approval rule (Modifica regola di approvazione), apportare le modifiche desiderate alla regola e quindi scegliere Submit (Invia).

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. Al termine della configurazione della regola di approvazione, scegliere Submit (Invia).

Modifica o elimina una regola di approvazione per una pull request (AWS CLI)

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

È possibile utilizzare il AWS CLI per modificare il contenuto di una regola di approvazione e per eliminare una regola di approvazione.

 Note

Potresti anche essere interessato ai seguenti comandi:

- [update-pull-request-approval-state](#), per approvare o revocare l'approvazione su una richiesta pull.
- [get-pull-request-approval-states](#), per visualizzare le approvazioni nella richiesta pull.
- [evaluate-pull-request-approval-rules](#), per stabilire se le condizioni per le regole di approvazione per una richiesta pull sono state soddisfatte.

Da utilizzare AWS CLI per modificare o eliminare una regola di approvazione per una pull request in un CodeCommit repository

1. Per modificare una regola di approvazione, eseguire il comando `update-pull-request-approval-rule-content` specificando:
 - L'ID della richiesta di pull (con l'opzione `--id`).
 - Il nome della regola di approvazione (con l'opzione `--approval-rule-name`).
 - Il contenuto della regola di approvazione (con l'opzione `--approval-rule-content`).

Questo esempio aggiorna una regola di approvazione denominata *Require two approved approvers* per una richiesta pull con l'ID di *27*. La regola richiede l'approvazione di un utente da un pool di approvazione che include qualsiasi utente IAM nell'account *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. Per eliminare una regola di approvazione, eseguire il comando `delete-pull-request-approval-rule`, specificando:
 - L'ID della richiesta di pull (con l'opzione `--id`).
 - Il nome della regola di approvazione (con l'opzione `--approval-rule-name`).

Ad esempio, per eliminare una regola di approvazione con il nome *My Approval Rule* di una pull request con l'ID di *15*:

```
aws codecommit delete-pull-request-approval-rule --pull-request-id 15 --approval-rule-name "My Approval Rule"
```

Sostituisci le regole di approvazione su una pull request

Nel corso normale dello sviluppo, si desidera che gli utenti soddisfino le condizioni delle regole di approvazione prima di unire le richieste pull. Tuttavia, in alcuni casi potrebbe essere necessario accelerare l'unione di una richiesta pull. Ad esempio, è potrebbe essere necessario inserire una correzione di bug in produzione, ma nessuno nel pool di approvazione è disponibile per approvare la richiesta pull. In casi come questi, è possibile scegliere di ignorare le regole di approvazione in una richiesta pull. È possibile ignorare tutte le regole di approvazione per una richiesta pull, incluse quelle create appositamente per la richiesta pull e generate da un modello di regola di approvazione. Non è possibile ignorare selettivamente una regola di approvazione specifica, ma solo tutte le regole. Dopo aver impostato i requisiti della regola di approvazione sovrascrivendo le regole, è possibile unire la richiesta pull al relativo ramo di destinazione.

Quando si sostituiscono le regole di approvazione in una richiesta pull, le informazioni sull'utente che ha ignorato le regole vengono registrate nell'attività per la richiesta pull. In questo modo puoi tornare indietro nella cronologia di una richiesta pull e rivedere chi ha ignorato le regole. Puoi anche scegliere di revocare l'override se la richiesta pull è ancora aperta. Dopo che la richiesta pull è stata unita, non è più possibile revocare l'override.

Argomenti

- [Sostituisci le regole di approvazione \(console\)](#)
- [Sostituisci le regole di approvazione \(\)AWS CLI](#)

Sostituisci le regole di approvazione (console)

È possibile ignorare i requisiti delle regole di approvazione in una richiesta pull nella console, come parte della revisione di una richiesta pull. Se si cambia idea, è possibile revocare la sostituzione e i requisiti della regola di approvazione vengono riapplicati. È possibile ignorare le regole di

approvazione o revocare una sostituzione solo se la richiesta pull è ancora aperta. Se viene unita o chiusa, non è possibile modificarne lo stato di sostituzione.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), scegliere il nome del repository.
3. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull). Scegliere la richiesta pull in cui si desidera sostituire i requisiti delle regole di approvazione o revocare una sostituzione.
4. Nella scheda Approvals (Approvazioni) scegliere Override approval rules (Sostituisci regole di approvazione). I requisiti vengono impostati e il testo del pulsante cambia in Revoke override (Revoca override). Per riapplicare i requisiti della regola di approvazione, scegliere Revoke override (Revoca sostituzione).

Sostituisci le regole di approvazione ()AWS CLI

È possibile utilizzare il AWS CLI per sovrascrivere i requisiti delle regole di approvazione. È inoltre possibile utilizzarlo per visualizzare lo stato di sostituzione di una richiesta pull.

Per sostituire i requisiti della regola di approvazione in una richiesta pull

1. Dal terminale o dalla riga di comando, eseguire il comando `override-pull-request-approval-rules`, specificando:
 - L'ID generato dal sistema della richiesta pull.
 - L'ultimo ID di revisione della richiesta pull. Per visualizzare queste informazioni, utilizzare `get-pull-request`.
 - Lo stato che si desidera per la sostituzione, `OVERRIDE` o `REVOKE`. Lo stato `REVOKE` rimuove lo stato `OVERRIDE`, ma non viene salvato.

Ad esempio, per sostituire le regole di approvazione su una richiesta pull con un ID di **34** e un ID di revisione di **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8EXAMPLE --override-status OVERRIDE
```

2. In caso di successo, questo comando non restituisce alcun risultato.
3. Per revocare la sostituzione su una richiesta pull con un ID di **34** e un ID di revisione di **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8dEXAMPLE --override-status REVOKE
```

Per ottenere informazioni sullo stato di sostituzione di una richiesta pull

1. Dal terminale o dalla riga di comando, eseguire il comando `get-pull-request-override-state`, specificando:
 - L'ID generato dal sistema della richiesta pull.
 - L'ultimo ID di revisione della richiesta pull. Per visualizzare queste informazioni, utilizzare `get-pull-request`.

Ad esempio, per visualizzare lo stato di sostituzione per una richiesta pull con un ID di **34** e un ID di revisione di **927df8d8dEXAMPLE**:

```
aws codecommit get-pull-request-override-state --pull-request-id 34 --revision-  
id 927df8d8dEXAMPLE
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "overridden": true,  
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

Unisci una richiesta pull in un repository AWS CodeCommit

Una volta esaminato il codice e soddisfatte tutte le regole di approvazione (se presenti) nella richiesta pull, è possibile unire una richiesta pull in uno dei diversi modi illustrati di seguito:

- È possibile utilizzare la console per unire il ramo di origine al ramo di destinazione utilizzando una delle strategie di unione disponibili, che chiude anche la richiesta di pull. È inoltre possibile risolvere qualsiasi conflitto di unione nella console. Nella console viene visualizzato un messaggio che indica se la richiesta pull può essere unita o se i conflitti devono essere risolti. Quando tutti i conflitti vengono risolti e si sceglie Merge (Unisci), l'unione viene eseguita utilizzando la strategia di

unione scelta. L'inoltro rapido è la strategia di unione predefinita, che è l'opzione predefinita per Git. A seconda dello stato del codice nei rami di origine e di destinazione, tale strategia potrebbe non essere disponibile, ma potrebbero esserne disponibili altre, ad esempio la strategia squash o quella a 3 vie.

- È possibile utilizzare la AWS CLI per unire e chiudere la pull request utilizzando la strategia fast-forward, squash o 3-way merge.
- Sul tuo computer locale, puoi usare il comando `git merge` per unire il ramo di origine nel ramo di destinazione e quindi eseguire il push del codice unito nel ramo di destinazione. Questo approccio presenta degli svantaggi da considerare attentamente. Unisce la richiesta di pull indipendentemente dal fatto che siano stati soddisfatti i requisiti per le regole di approvazione della richiesta di pull, eludendo tali controlli. L'unione e il push del ramo di destinazione chiudono automaticamente anche la richiesta pull, se la richiesta pull viene unita utilizzando la strategia di unione inoltro rapido. Un vantaggio di questo approccio è che il `git merge` comando consente di scegliere opzioni o strategie di unione che non sono disponibili nella console. CodeCommit Per ulteriori informazioni su `git merge` e sulle opzioni di unione, consulta [git-merge](#) o la documentazione Git.

CodeCommit chiude automaticamente una richiesta pull se il ramo di origine o di destinazione della pull request viene eliminato.

Argomenti

- [Unisce una pull request \(console\)](#)
- [Unisci una richiesta pull \(\)AWS CLI](#)

Unisce una pull request (console)

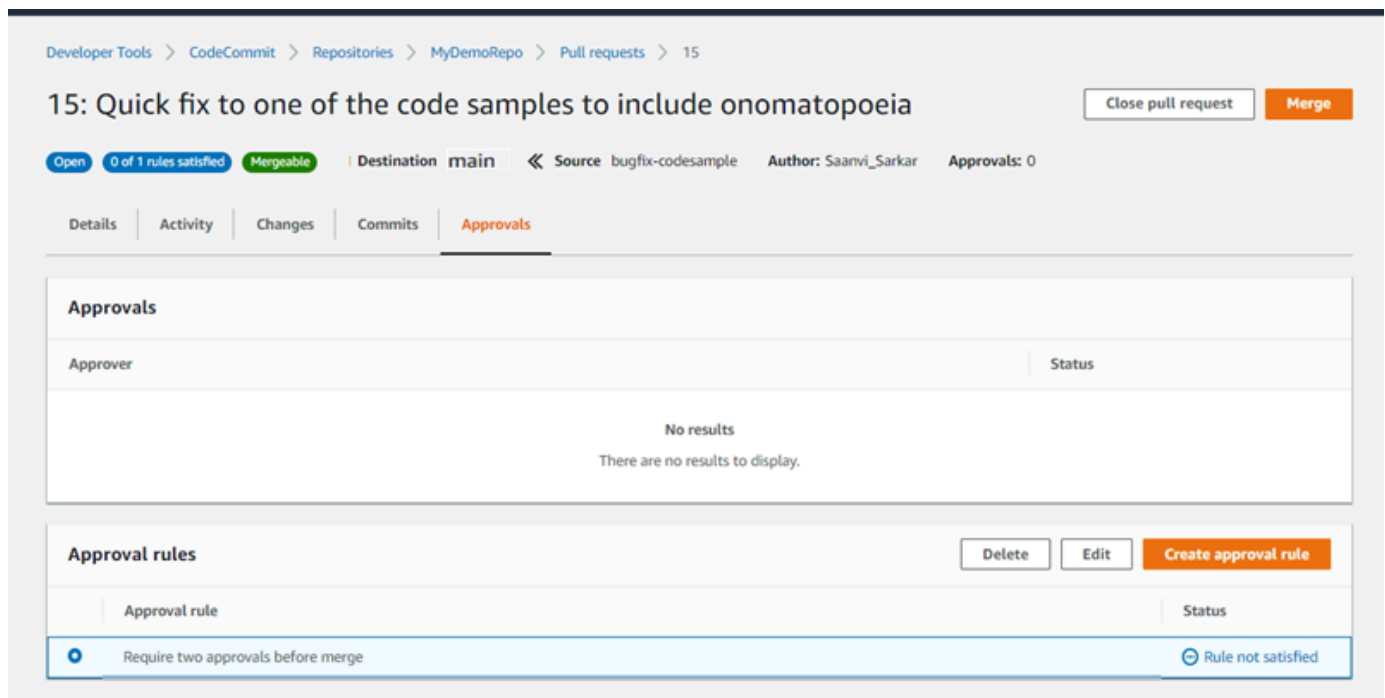
È possibile utilizzare la CodeCommit console per unire una richiesta pull in un CodeCommit repository. Una volta modificato in Merged (Unito), lo stato di una richiesta di pull non appare più nell'elenco delle richieste di pull aperte. Una richiesta di pull unita è categorizzata come chiusa. Non può essere modificato in Open (Apri), ma gli utenti possono comunque commentare le modifiche e rispondere ai commenti. Dopo l'unione o la chiusura di una richiesta pull, non è possibile approvarla, revocare l'approvazione o ignorare le regole di approvazione applicate alla richiesta pull.

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories (Repository), scegliere il nome del repository.

3. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull).
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte. Scegliere la richiesta di pull aperta che si desidera unire.
5. Nella richiesta pull, scegliere Approvals (Approvazioni). Esaminare l'elenco degli approvatori e verificare che siano state soddisfatte le condizioni per tutte le regole di approvazione (se presenti). Non è possibile unire una richiesta pull se una o più regole di approvazione hanno lo stato Rule not satisfied (Regola non soddisfatta). Se nessuno ha approvato la richiesta pull, valutare se si desidera unirla o se si desidera attendere le approvazioni.

Note

Se è stata creata una regola di approvazione per una richiesta pull, è possibile modificarla o eliminarla per sbloccare l'unione. Se la regola di approvazione è stata creata con un modello di regola di approvazione, non è possibile modificarla o eliminarla. È possibile scegliere solo di ignorare i requisiti. Per ulteriori informazioni, consulta [Sostituisci le regole di approvazione su una pull request](#).



The screenshot shows the AWS CodeCommit Pull Request interface for a pull request titled "15: Quick fix to one of the code samples to include onomatopoeia". The interface includes a breadcrumb trail: Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 15. At the top right, there are buttons for "Close pull request" and "Merge". Below the title, there are status indicators: "Open", "0 of 1 rules satisfied", and "Mergeable". The "Destination" is set to "main", and the "Source" is "bugfix-codesample". The "Author" is "Saarvi_Sarkar" and "Approvals" are 0. The "Approvals" tab is selected, showing a table with columns "Approver" and "Status". The table is empty, with a message "No results. There are no results to display." Below the table, there is a section for "Approval rules" with buttons for "Delete", "Edit", and "Create approval rule". A table lists the approval rules, with one rule: "Require two approvals before merge" with a status of "Rule not satisfied".

6. Selezionare Merge (Unisci).
7. Nella richiesta di pull, scegliere tra le strategie di unione disponibili. Le strategie di unione non applicabili vengono visualizzate in grigio. Se non sono disponibili strategie di unione, puoi

scegliere di risolvere manualmente i conflitti nella CodeCommit console oppure puoi risolverli localmente usando il tuo client Git. Per ulteriori informazioni, consulta [Risolvi i conflitti in una pull request in un AWS CodeCommit repository](#).

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 unione ad avanzamento rapido farà avanzare il riferimento per il ramo di destinazione al più recente commit del ramo di origine. Questo è il comportamento predefinito di Git quando possibile. Non viene creato nessun commit di unione, ma tutta la cronologia di commit dal ramo di origine viene conservata come se fosse avvenuta nel ramo di destinazione. Le unioni di avanzamento rapido non vengono visualizzate come unione di ramo nella visualizzazione

del commit della cronologia del ramo di destinazione perché non viene creato alcun commit di unione. La punta del ramo di origine viene inoltrata rapidamente alla punta del ramo di destinazione.

- Una unione squash crea un commit contenente le modifiche nel ramo di origine e applica quel singolo commit squash al ramo di destinazione. Per impostazione predefinita, il messaggio di commit per il commit squash contiene tutti i messaggi di commit delle modifiche apportate al ramo di origine. Non verrà conservata nessuna cronologia di commit singola delle modifiche del ramo. Questo può aiutare a semplificare la cronologia dei repository conservando una rappresentazione grafica dell'unione nella vista del visualizzatore di commit della cronologia dei rami di destinazione.
 - Una unione a tre vie creerà un commit di unione per l'unione nel ramo di destinazione, ma conserverà anche i commit singoli effettuati nel ramo di origine come parte della cronologia del ramo di destinazione. Questo può aiutare a mantenere una cronologia completa delle modifiche al repository.
8. Se si sceglie la strategia di unione squash o a tre vie, rivedere il messaggio di commit generato automaticamente e modificarlo se si desidera cambiare le informazioni. Aggiungere il nome e l'indirizzo e-mail per la cronologia di commit.
 9. (Facoltativo) Deselezionare l'opzione per eliminare il ramo di origine come parte dell'unione. L'impostazione predefinita è eliminare il ramo di origine quando viene unita una richiesta pull.
 10. Scegliere Merge pull request (Unisci richiesta di pull) per completare l'unione.

Unisci una richiesta pull ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Da utilizzare AWS CLI per unire le richieste pull in un repository CodeCommit

1. Per valutare se per una richiesta pull sono state soddisfatte tutte le regole di approvazione e risulta essere pronta per l'unione, eseguire il comando `evaluate-pull-request-approval-rules`, specificando:
 - L'ID della richiesta di pull (da utilizzare con l'opzione `--pull-request-id`).
 - L'ID di revisione della richiesta pull, utilizzando il `--revision-id` option). È possibile ottenere l'ID di revisione corrente per una richiesta pull utilizzando il comando [get-pull-request](#).

Ad esempio, per valutare lo stato delle regole di approvazione su una pull request con un ID **27** e un ID di revisione pari a: **9f29d167EXAMPLE**

```
aws codecommit evaluate-pull-request-approval-rules --pull-request-id 27 --  
revision-id 9f29d167EXAMPLE
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

Note

Questo output indica che una richiesta pull non può essere unita perché i requisiti di una regola di approvazione non sono stati soddisfatti. Per unire questa richiesta pull, è possibile chiedere ai revisori di approvarla per soddisfare le condizioni della regola. A seconda delle autorizzazioni e della modalità di creazione della regola, è possibile modificare, ignorare o eliminare la regola. Per ulteriori informazioni, consulta [Revisione di una richiesta di pull](#), [Sostituisci le regole di approvazione su una pull request](#) e [Modifica o elimina una regola di approvazione per una pull request](#).

2. Per unire e chiudere una richiesta di pull utilizzando una strategia di unione di avanzamento rapido, eseguire il comando `merge-pull-request-by-fast-forward`, specificando:
 - L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
 - L'ID commit completo dell'estremità del ramo di origine (con l'opzione `--source-commit-id`).
 - Il nome del repository (con l'opzione `--repository-name`).

Ad esempio, per unire e chiudere una richiesta pull con l'ID di **47** e un ID di commit di origine **99132ab0EXAMPLE** in un repository denominato: **MyDemoRepo**

```
aws codecommit merge-pull-request-by-fast-forward --pull-request-id 47 --source-commit-id 99132ab0EXAMPLE --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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. Per unire e chiudere una richiesta di pull utilizzando la strategia di unione squash, eseguire il comando `merge-pull-request-by-squash`, specificando:
 - L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
 - L'ID commit completo dell'estremità del ramo di origine (con l'opzione `--source-commit-id`).
 - Il nome del repository (con l'opzione `--repository-name`).
 - Il livello di dettaglio del conflitto che si desidera utilizzare (con l'opzione `--conflict-detail-level`). Se non altrimenti specificato, si utilizza l'impostazione predefinita **FILE_LEVEL**.
 - La strategia di risoluzione dei conflitti che si desidera utilizzare (con l'opzione `--conflict-resolution-strategy`). Se non altrimenti specificato, l'impostazione predefinita è **NONE** e i conflitti devono essere risolti manualmente.
 - Il messaggio di commit da includere (con l'opzione `--commit-message`).
 - Il nome da utilizzare per il commit (con l'opzione `--author-name`).
 - L'indirizzo e-mail da utilizzare per il commit (con l'opzione `--email`).
 - Per mantenere le cartelle vuote (con l'opzione `--keep-empty-folders`).

L'esempio seguente unisce e chiude una richiesta pull con l'ID di **47** e l'ID di commit di origine **99132ab0EXAMPLE** in un repository denominato **MyDemoRepo**. Utilizza i dettagli del conflitto di **LINE_LEVEL** e la strategia di risoluzione dei conflitti di **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"

```

In caso di successo, questo comando produce lo stesso tipo di output dell'unione mediante l'avanzamento rapido, output simile ai seguenti:

```
{
```

```

    "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. Per unire e chiudere una richiesta di pull utilizzando la strategia di unione a tre vie, eseguire il comando `merge-pull-request-by-three-way`, specificando:
- L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
 - L'ID commit completo dell'estremità del ramo di origine (con l'opzione `--source-commit-id`).
 - Il nome del repository (con l'opzione `--repository-name`).
 - Il livello di dettaglio del conflitto che si desidera utilizzare (con l'opzione `--conflict-detail-level`). Se non altrimenti specificato, si utilizza l'impostazione predefinita **FILE_LEVEL**.
 - La strategia di risoluzione dei conflitti che si desidera utilizzare (con l'opzione `--conflict-resolution-strategy`). Se non altrimenti specificato, l'impostazione predefinita è **NONE** e i conflitti devono essere risolti manualmente.
 - Il messaggio di commit da includere (con l'opzione `--commit-message`).
 - Il nome da utilizzare per il commit (con l'opzione `--author-name`).
 - L'indirizzo e-mail da utilizzare per il commit (con l'opzione `--email`).
 - Per mantenere le cartelle vuote (con l'opzione `--keep-empty-folders`).

L'esempio seguente unisce e chiude una pull request con l'ID di **47** e un source commit ID di **99132ab0EXAMPLE** in un repository denominato **MyDemoRepo**. Utilizza le opzioni predefinite per i dettagli dei conflitti e la strategia di risoluzione dei conflitti:

```
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"
```

In caso di successo, questo comando produce lo stesso tipo di output dell'unione mediante l'avanzamento rapido, simile ai seguenti:

```
{
  "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": 571356106.936,
        "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"
}

```

Risolvi i conflitti in una pull request in un AWS CodeCommit repository

Se la richiesta di pull presenta dei conflitti e non può essere unita, è possibile provare a risolvere i conflitti in uno dei seguenti modi:

- Sul computer locale, è possibile utilizzare il comando `git diff` per trovare i conflitti tra i due rami e apportare modifiche per risolverli. È inoltre possibile utilizzare uno strumento delle differenze o altri prodotti software per aiutare i clienti a individuare e risolvere le differenze. Dopo averli risolti in modo soddisfacente, puoi inviare al ramo sorgente le modifiche che contengono i conflitti risolti, aggiornando così la pull request. Per ulteriori informazioni su `git diff` e `git difftool`, consulta la documentazione di Git.
- Nella console, è possibile scegliere **Resolve conflicts (Risolvi conflitti)**. Si apre un editor di testo normale che mostra i conflitti in modo analogo al comando `git diff`. È possibile rivedere manualmente i conflitti in ogni file che li contiene, apportare le modifiche e quindi aggiornare la richiesta con le modifiche apportate.
- In AWS CLI, puoi usare il AWS CLI per ottenere informazioni sui conflitti di unione e creare un commit di unione senza riferimenti per testare un'unione.

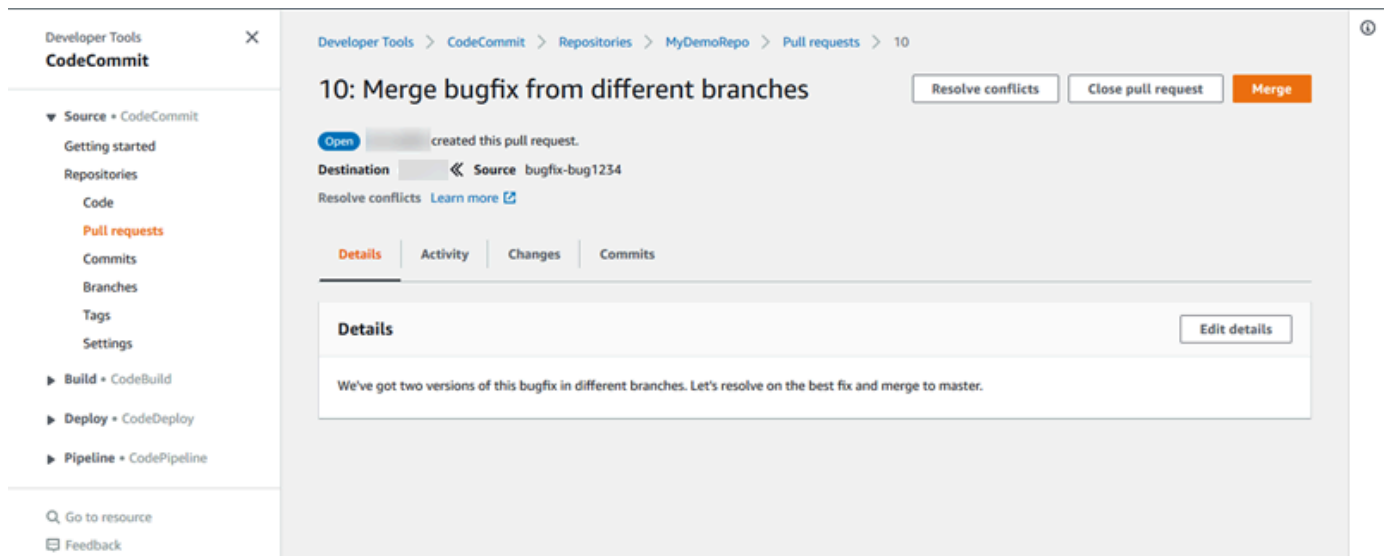
Argomenti

- [Risolvi i conflitti in una pull request \(console\)](#)
- [Risolvi i conflitti in una pull request \(\)AWS CLI](#)

Risolvi i conflitti in una pull request (console)

È possibile utilizzare la CodeCommit console per risolvere i conflitti in una richiesta pull in un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In **Repositories (Repository)**, scegliere il nome del repository.
3. Nel riquadro di navigazione, selezionare **Pull requests (Richieste pull)**.
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte. Scegliere la richiesta di pull aperta da unire ma questa contiene i conflitti.
5. Nella richiesta di pull, scegliere **Resolve conflicts (Risolvi conflitti)**. Questa opzione è disponibile solo se non ci sono conflitti che devono essere risolti prima che la richiesta di pull possa essere unita.



6. Si apre una finestra di risoluzione dei conflitti che mostra un elenco di tutti i file che presentano conflitti da risolvere. Scegliere ogni file nell'elenco per rivedere i conflitti e apportare le modifiche necessarie finché tutti i conflitti saranno stati risolti.

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

- È possibile scegliere di utilizzare i contenuti del file di origine, i contenuti del file di destinazione oppure se il file non è un file binario, per modificare manualmente i contenuti di un file in modo che contenga solo le modifiche desiderate. I marker git diff standard vengono utilizzati per visualizzare i conflitti tra la destinazione (HEAD) e i rami di origine nel file.
- Se un file è un file binario, un modulo secondario Git oppure se esiste un conflitto di nome file/cartella, è necessario scegliere di utilizzare il file di origine o il file di destinazione per risolvere i conflitti. Non è possibile visualizzare o modificare file binari nella console. CodeCommit
- In caso di conflitti tra le modalità di file, viene visualizzata l'opzione per risolvere tale conflitto scegliendo tra la modalità di file del file di origine e la modalità di file del file di destinazione.

- Se si decide di ignorare le modifiche per un file e ripristinarne lo stato in conflitto, scegliere **Reset file (Ripristina file)**. In questo modo sarà possibile risolvere i conflitti in modo diverso.
7. Se si è soddisfatti delle modifiche, scegliere **Update pull request (Aggiorna pull di richiesta)**.

Note

È necessario risolvere tutti i conflitti in tutti i file prima di aggiornare con successo la richiesta di pull con le modifiche apportate.

8. La richiesta di pull viene aggiornata con le modifiche e può essere unita. Viene visualizzata la pagina di unione. È possibile scegliere di unire la richiesta di pull in questo momento oppure è possibile tornare all'elenco delle richieste di pull.

Risolvi i conflitti in una pull request ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Nessun AWS CLI comando consente di risolvere i conflitti in una richiesta pull e di unire tale richiesta. Tuttavia, è possibile utilizzare i comandi individuali per individuare i conflitti, tentare di risolverli e verificare se una richiesta di pull può essere unita. È possibile utilizzare:

- `get-merge-options`, per conoscere quali opzioni di unione sono disponibili per una unione tra due specificatori di commit.
- `get-merge-conflicts`, per restituire un elenco di file con conflitti di unione in un'unione tra due specificatori di commit.
- `batch-describe-merge-conflicts`, per ottenere informazioni su tutti i conflitti di unione nei file in un'unione tra due commit utilizzando una strategia di unione specificata.
- `describe-merge-conflicts`, per ottenere informazioni dettagliate su conflitti di unione per un determinato file tra due commit utilizzando una strategia di unione specificata.
- `create-unreferenced-merge-commit`, per testare il risultato dell'unione di due specificatori di commit utilizzando una strategia di unione specificata.

- 1.
- Per scoprire quali opzioni di unione sono disponibili per una fusione tra due specificatori di commit, eseguire il comando `get-merge-options`, specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- Il nome del repository (con l'opzione `--repository-name`).
- (Facoltativo) Una strategia di risoluzione dei conflitti da utilizzare (con l'opzione `--conflict-resolution-strategy`).
- (Facoltativo) Il livello di dettaglio delle informazioni sui conflitti (con l'opzione `--conflict-detail-level`).

Ad esempio, per determinare le strategie di unione disponibili per unire un ramo di origine denominato *bugfix-1234* con un ramo di destinazione denominato in un repository denominato: *main MyDemoRepo*

```
aws codecommit get-merge-options --source-commit-specifier bugfix-1234 --  
destination-commit-specifier main --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "mergeOptions": [  
    "FAST_FORWARD_MERGE",  
    "SQUASH_MERGE",  
    "THREE_WAY_MERGE"  
  ],  
  "sourceCommitId": "d49940adEXAMPLE",  
  "destinationCommitId": "86958e0aEXAMPLE",  
  "baseCommitId": "86958e0aEXAMPLE"  
}
```

2.

Per ottenere un elenco dei file contenenti i conflitti di unione in un'unione tra due specificatori di commit, eseguire il comando `get-merge-conflicts`, specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- Il nome del repository (con l'opzione `--repository-name`).

- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).
- (Facoltativo) Il livello di dettaglio delle informazioni sui conflitti (con l'opzione `--conflict-detail-level`).
- (Facoltativo) Una strategia di risoluzione dei conflitti da utilizzare (con l'opzione `--conflict-resolution-strategy`).
- (Facoltativo) Il numero massimo di file con conflitti da restituire (con l'opzione `--max-conflict-files`).

Ad esempio, per ottenere un elenco di file che contengono conflitti in seguito a un'unione tra un ramo di origine denominato `feature-randomizationfeature` e un ramo di destinazione denominato `main`, utilizza la strategia di unione a tre vie in un repository denominato: `MyDemoRepo`

```
aws codecommit get-merge-conflicts --source-commit-specifier feature-
randomizationfeature --destination-commit-specifier main --merge-option
THREE_WAY_MERGE --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.

Per ottenere informazioni sui conflitti di unione in tutti i file o in un sottoinsieme di file in una unione tra due specificatori di commit, eseguire il comando `batch-describe-merge-conflicts`, specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).
- Il nome del repository (con l'opzione `--repository-name`).
- (Facoltativo) Una strategia di risoluzione dei conflitti da utilizzare (con l'opzione `--conflict-resolution-strategy`).
- (Facoltativo) Il livello di dettaglio delle informazioni sui conflitti (con l'opzione `--conflict-detail-level`).
- (Facoltativo) Il numero massimo di hunk di unione da restituire (con l'opzione `--max-merge-hunks`).
- (Facoltativo) Il numero massimo di file con conflitti da restituire (con l'opzione `--max-conflict-files`).
- (Facoltativo) il percorso dei file di destinazione da utilizzare per descrivere i conflitti (con l'opzione `--file-paths`).

Ad esempio, per determinare i conflitti di unione dovuti all'unione di un ramo di origine denominato *feature-randomizationfeature* con un ramo di destinazione denominato utilizzando la strategia in un repository denominato: *main THREE_WAY_MERGE MyDemoRepo*

```
aws codecommit batch-describe-merge-conflicts --source-commit-specifier feature-randomizationfeature --destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.

Per ottenere informazioni dettagliate su qualsiasi conflitto di unione per un determinato file in un'unione tra due specificatori di commit, eseguire il comando `describe-merge-conflicts`, specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).
- Il percorso dei file di destinazione da utilizzare per descrivere i conflitti (con l'opzione `--file-path`).
- Il nome del repository (con l'opzione `--repository-name`).
- (Facoltativo) Una strategia di risoluzione dei conflitti da utilizzare (con l'opzione `--conflict-resolution-strategy`).

- (Facoltativo) Il livello di dettaglio delle informazioni sui conflitti (con l'opzione `--conflict-detail-level`).
- (Facoltativo) Il numero massimo di hunk di unione da restituire (con l'opzione `--max-merge-hunks`).
- (Facoltativo) Il numero massimo di file con conflitti da restituire (con l'opzione `--max-conflict-files`).

Ad esempio, per determinare i conflitti di unione per un file denominato *readme.md* in un ramo di origine denominato *feature-randomizationfeature* con un ramo di destinazione denominato *main* utilizzando la *THREE_WAY_MERGE* strategia in un repository denominato: *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
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.

Per creare un commit non referenziato che rappresenta il risultato dell'unione di due specificatori di commit, eseguire il comando `create-unreferenced-merge-commit`, specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).
- Il nome del repository (con l'opzione `--repository-name`).
- (Facoltativo) Una strategia di risoluzione dei conflitti da utilizzare (con l'opzione `--conflict-resolution-strategy`).

- (Facoltativo) Il livello di dettaglio delle informazioni sui conflitti (con l'opzione `--conflict-detail-level`).
- (Facoltativo) Il messaggio di commit da includere (con l'opzione `--commit-message`).
- (Facoltativo) Il nome da utilizzare per il commit (con l'opzione `--name`).
- (Facoltativo) L'indirizzo e-mail da utilizzare per il commit (con l'opzione `--email`).
- (Facoltativo) Per mantenere le cartelle vuote (con l'opzione `--keep-empty-folders`).

Ad esempio, per determinare i conflitti di unione per l'unione di un ramo di origine denominato *bugfix-1234* con un ramo di destinazione denominato *main* utilizzando la strategia `ACCEPT_SOURCE` in un repository denominato: *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."
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Chiudere una pull request in un repository AWS CodeCommit

Se si desidera chiudere una richiesta di pull senza unire il codice, è possibile farlo in uno dei seguenti modi:

- Nella console, puoi chiudere una richiesta di pull senza unire il codice. Potresti voler eseguire questa operazione quando vuoi utilizzare il comando `git merge` per unire manualmente i rami o se il codice nel ramo di origine della richiesta di pull non è il codice che vuoi unire al ramo di destinazione.
- È possibile eliminare il ramo di origine specificato nella pull request. CodeCommit chiude automaticamente una richiesta pull se il ramo di origine o di destinazione della pull request viene eliminato.

- In AWS CLI, puoi aggiornare lo stato di una pull request da OPEN a CLOSED. Questo chiude la richiesta di pull senza unire il codice.

Argomenti

- [Chiudi una pull request \(console\)](#)
- [Chiudere una pull request \(AWS CLI\)](#)

Chiudi una pull request (console)

È possibile utilizzare la CodeCommit console per chiudere una richiesta pull in un CodeCommit repository. Una volta che lo stato di una richiesta di pull è diventato Closed (Chiuso), non può tornare Open (Aperto), ma gli utenti possono ancora commentare le modifiche e rispondere ai commenti.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), scegliere il nome del repository.
3. Nel riquadro di navigazione, selezionare Pull requests (Richieste pull).
4. Per impostazione predefinita, viene visualizzato l'elenco di tutte le richieste pull aperte. Scegliere la richiesta di pull aperta che si desidera chiudere.

Pull requests

Info

Q

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

5. Nella richiesta di pull, scegliere Close pull request (Chiudi richiesta di pull). Questa opzione chiude la richiesta di pull senza tentare di unire il ramo di origine nel ramo di destinazione. Questa opzione non fornisce un modo per eliminare il ramo di origine come parte della chiusura della richiesta di pull, ma è possibile farlo dopo che la richiesta è stata chiusa.

Chiudere una pull request (AWS CLI)

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Da utilizzare AWS CLI per chiudere le richieste pull in un CodeCommit repository

- Per aggiornare lo stato di una richiesta di pull in un repository da OPEN a CLOSED, eseguire il comando `update-pull-request-status` specificando:
 - L'ID della richiesta di pull (con l'opzione `--pull-request-id`).
 - Lo stato della richiesta di pull (con l'opzione `--pull-request-status`).

Ad esempio, per aggiornare lo stato di una pull request con l'ID di **42** a uno status di **CLOSED** in un CodeCommit repository denominato: MyDemoRepo

```
aws codecommit update-pull-request-status --pull-request-id 42 --pull-request-status CLOSED
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codecommit:GetRepository",
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ]
    }
  ]
}
```

Utilizzo dei modelli di regola di approvazione

È possibile creare regole di approvazione per le richieste pull. Per applicare automaticamente le regole di approvazione ad alcune o tutte le richieste pull create nei repository, utilizza i modelli di regole di approvazione. I modelli di regola di approvazione consentono di personalizzare i flussi di lavoro di sviluppo tra repository in modo che i diversi rami abbiano livelli appropriati di approvazione e controllo. È possibile definire regole diverse per i rami di produzione e sviluppo. Tali regole vengono applicate ogni qualvolta venga creata una richiesta pull corrispondente alle condizioni della regola. Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, consulta [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#)

È possibile associare un modello di regola di approvazione a uno o più repository nel luogo in Regione AWS cui vengono creati. Se associato a un repository, un modello crea automaticamente regole di approvazione per le richieste pull in tale repository come parte della creazione della richiesta pull. Proprio come una singola regola di approvazione, un modello di regola di approvazione definisce una struttura di regole di approvazione, inclusi il numero di approvazioni richieste e un pool facoltativo di utenti da cui devono provenire le approvazioni. A differenza di una regola di approvazione, è anche possibile definire i riferimenti di destinazione (il ramo o i rami), noti anche come filtri di ramo. Se si definiscono i riferimenti di destinazione, solo le richieste pull, i cui nomi di ramo di destinazione corrispondono ai nomi di ramo specificati (riferimenti di destinazione) nel modello, sono create per tali regole. Ad esempio, se si specifica **refs/heads/main** come riferimento di destinazione, la regola di approvazione definita nel modello viene applicata solo alle richieste pull se il ramo di destinazione è **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 ✕

Argomenti

- [Creare un modello di regola di approvazione](#)
- [Associa un modello di regola di approvazione a un repository](#)
- [Gestire i modelli di regola di approvazione](#)
- [Dissociare un modello di regola di approvazione](#)
- [Eliminare un modello di regola di approvazione](#)

Creare un modello di regola di approvazione

È possibile creare uno o più modelli di regola di approvazione per personalizzare i flussi di lavoro di sviluppo tra repository. Creando più modelli, è possibile configurare la creazione automatica delle regole di approvazione in modo che i diversi rami abbiano livelli appropriati di approvazioni e controllo. Ad esempio, è possibile creare modelli diversi per i rami di produzione e sviluppo e applicare questi modelli a uno o più repository. Quando gli utenti creano richieste pull in tali repository, la richiesta viene valutata rispetto a tali modelli. Se la richiesta corrisponde alle condizioni nei modelli applicati, vengono create regole di approvazione per la richiesta pull.

È possibile utilizzare la console o creare modelli AWS CLI di regole di approvazione. Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, consulta [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#).

Argomenti

- [Creare un modello di regola di approvazione \(console\)](#)
- [Crea un modello di regola di approvazione \(\)AWS CLI](#)

Creare un modello di regola di approvazione (console)

I modelli di regola di approvazione non sono associati a nessun repository per impostazione predefinita. È possibile creare un'associazione tra un modello e uno o più repository quando si crea il modello oppure è possibile aggiungere le associazioni in un secondo momento.

Per creare un modello di regola di approvazione (Console)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Scegliere Approval rule templates (Modelli di regola di approvazione), quindi Create template (Crea modello).

3. Nel Approval rule template name (Nome del modello di regola di approvazione), assegnare al modello un nome descrittivo in modo da sapere a cosa serve. Ad esempio, se si desidera richiedere a una persona di un gruppo di sviluppatori senior di approvare una richiesta pull prima che possa essere unita, è possibile assegnare un nome alla regola **Require 1 approver from a senior developer**.
4. (Facoltativo) In Description (Descrizione), fornire una descrizione dello scopo di questo modello. Questo può aiutare gli altri a decidere se questo modello è appropriato per i loro repository.
5. In Number of approvals needed (Numero di approvazioni necessarie), immettere il numero desiderato. Il valore di default è 1.
6. (Facoltativo) Se si desidera richiedere che le approvazioni per una richiesta pull provengano da un gruppo specifico di utenti, in Approval rule members (Membri della regola di approvazione), scegliere Add (Aggiungi). In Approver type (Tipo approvatore), scegliere una delle seguenti opzioni:
 - Nome utente IAM o ruolo assunto: questa opzione precompila l'ID dell'account Amazon Web Services per l'account utilizzato per accedere e richiede solo un nome. Può essere utilizzato sia per gli utenti IAM che per gli utenti ad accesso federato il cui nome corrisponde al nome fornito. Questa è un'opzione molto potente che offre una grande flessibilità. Ad esempio, se scegli questa opzione e hai effettuato l'accesso con l'account Amazon Web Services 123456789012 e specifichi **Mary_Major**, tutte le seguenti operazioni vengono conteggiate come approvazioni provenienti da quell'utente:
 - Un utente IAM nell'account () `arn:aws:iam::123456789012:user/Mary_Major`
 - Un utente federato identificato in IAM come Mary_Major ()
`arn:aws:sts::123456789012:federated-user/Mary_Major`

Questa opzione non riconosce una sessione attiva di qualcuno che assume il ruolo di **CodeCommitReview** con un nome di sessione di ruolo Mary_Major (`arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major`) a meno che non si includa un carattere jolly (`*Mary_Major`). È inoltre possibile specificare esplicitamente il nome del ruolo (`CodeCommitReview/Mary_Major`).

 - ARN completo: questa opzione consente di specificare l'Amazon Resource Name (ARN) completo dell'utente o del ruolo IAM. Questa opzione supporta anche i ruoli presunti utilizzati da altri AWS servizi, come e AWS Lambda . AWS CodeBuild Per i ruoli assunti, il formato ARN deve essere `arn:aws:sts::AccountID:assumed-role/RoleName` per i ruoli e `arn:aws:sts::AccountID:assumed-role/FunctionName` per le funzioni.

Se hai scelto il nome utente IAM o hai assunto il ruolo come tipo di approvatore, in Valore inserisci il nome dell'utente o del ruolo IAM o l'ARN completo dell'utente o del ruolo. Scegliere Add (Aggiungi) di nuovo per aggiungere altri utenti o ruoli, fino a quando non saranno stati aggiunti tutti gli utenti o i ruoli le cui approvazioni valgono per il numero di approvazioni richieste.

Entrambi i tipi di approvatore consentono di utilizzare caratteri jolly (*) nei loro valori. Ad esempio, se scegli l'opzione nome utente IAM o ruolo assunto e lo specifichi **CodeCommitReview/***, tutti gli utenti che assumono il ruolo di **CodeCommitReview** vengono conteggiati nel pool di approvazione. I nomi delle sessioni di ruolo individuali valgono per il numero richiesto di approvatori. In questo modo, sia Mary_Major sia Li_Juan contano come approvazioni quando accedono e assumono il ruolo di CodeCommitReview. Per ulteriori informazioni su IAM ARNs, wildcard e formati, consulta [IAM Identifiers](#).

 Note

Le regole di approvazione non supportano le approvazioni cross-account.

7. (Facoltativo) In Branch filters (Filtri ramo), immettere i nomi dei rami di destinazione da utilizzare per filtrare la creazione delle regole di approvazione. Ad esempio, se si specifica *main*, viene creata una regola di approvazione per le richieste pull nei repository associati solo se il ramo di destinazione per la pull request è un ramo denominato *main*. È possibile utilizzare i caratteri jolly (*) nei nomi dei rami per applicare le regole di approvazione a tutti i nomi di ramo che corrispondono ai caratteri jolly. Tuttavia, non è possibile utilizzare un carattere jolly all'inizio del nome di un ramo. È possibile specificare fino a 100 nomi di ramo. Se non si specificano filtri, il modello si applica a tutti i rami in un repository associato.
8. (Facoltativo) In Archivi associati, nell'elenco Repository, scegliete i repository in esso contenuti Regione AWS che desiderate associare a questa regola di approvazione.

 Note

È possibile scegliere di associare repository dopo aver creato il modello. Per ulteriori informazioni, consulta [Associa un modello di regola di approvazione a un repository](#).

9. Scegli Create (Crea) .

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 ✕

Crea un modello di regola di approvazione ()AWS CLI

È possibile utilizzare il AWS CLI per creare modelli di regole di approvazione. Quando si utilizza il AWS CLI, è possibile specificare i riferimenti di destinazione per il modello, in modo che si applichi solo alle richieste pull i cui rami di destinazione corrispondono a quelli del modello.

Per creare un modello di regola di approvazione (AWS CLI)

1. Dal terminale o dalla riga di comando, eseguire il comando `create-approval-rule-template`, specificando:
 - Nome per il modello di regola di approvazione. Prendere in considerazione l'utilizzo di un nome che ne descrive lo scopo.
 - Descrizione del modello di regola di approvazione. Come per il nome, prendere in considerazione la possibilità di fornire una descrizione dettagliata.
 - La struttura JSON del modello di regola di approvazione. Questa struttura può includere i requisiti per i riferimenti di destinazione, ovvero i rami di destinazione per le richieste pull a cui viene applicata la regola di approvazione, e i membri del pool di approvazione, che sono utenti le cui approvazioni contano per il numero di approvazioni richieste.

Quando si crea il contenuto della regola di approvazione, è possibile specificare gli approvatori in un pool di approvazioni in due modi:

- **CodeCommitApprovers**: questa opzione richiede solo un account Amazon Web Services e una risorsa. Può essere utilizzata sia per gli utenti IAM che per gli utenti ad accesso federato il cui nome corrisponde al nome della risorsa fornito. Questa è un'opzione molto potente che offre una grande flessibilità. Ad esempio, se si specifica l' AWS account 123456789012 e **Mary_Major** tutte le seguenti approvazioni vengono conteggiate come approvazioni provenienti da quell'utente:
 - Un utente IAM nell'account () `arn:aws:iam::123456789012:user/Mary_Major`
 - Un utente federato identificato in IAM come **Mary_Major** ()
`arn:aws:sts::123456789012:federated-user/Mary_Major`

Questa opzione non riconosce una sessione attiva di qualcuno che assume il ruolo di *SeniorDevelopers* con un nome di sessione di ruolo uguale a () a meno che non si includa un wildcard *Mary_Major* (`arn:aws:sts::123456789012:assumed-role/SeniorDevelopers/Mary_Major`). ***Mary_Major**

- ARN completo: questa opzione consente di specificare l'Amazon Resource Name (ARN) completo dell'utente o del ruolo IAM.

[Per ulteriori informazioni su IAM ARNs, i wildcard e i formati, consulta IAM Identifiers.](#)

Nell'esempio seguente viene creato un modello di regola di approvazione denominato **2-approver-rule-for-main** e una descrizione di **Requires two developers from the team to approve the pull request if the destination branch is main**. Il modello richiede due utenti che assumono il ruolo di **CodeCommitReview** per approvare qualsiasi richiesta pull prima che possa essere unita al ramo **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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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"
  }
}
```

Associa un modello di regola di approvazione a un repository

I modelli di regole di approvazione vengono creati in uno specifico Regione AWS, ma non influiscono sui repository in esso contenuti Regione AWS finché non vengono associati. Per applicare un modello a uno o più repository, è necessario associare il modello al repository o ai repository. È possibile applicare un singolo modello a più repository in un unico. Regione AWS Ciò consente di automatizzare e standardizzare il flusso di lavoro di sviluppo nei repository, creando condizioni coerenti per l'approvazione e l'unione delle richieste pull.

È possibile associare un modello di regola di approvazione solo ai repository in Regione AWS cui è stato creato il modello di regola di approvazione.

Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, vedere [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#)

Argomenti

- [Associare un modello di regola di approvazione \(console\)](#)
- [Associa un modello di regola di approvazione \(\)AWS CLI](#)

Associare un modello di regola di approvazione (console)

È possibile che tu abbia associato i repository a un modello di regola di approvazione al momento della creazione. Questa operazione è facoltativa. È possibile aggiungere o rimuovere associazioni modificando il modello.

Per associare un modello di regola di approvazione ai repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Scegliere i Approval rule templates (Modelli di regola di approvazione). Scegliere il modello, quindi scegliere Edit (Modifica).
3. In Associated Repositories (Repository associati), scegliere i repository dall'elenco Repositories (Repositories). Ogni repository associato appare sotto la casella di elenco.
4. Seleziona Salva. Le regole di approvazione vengono ora applicate a tutte le richieste pull create nei repository associati.

Associa un modello di regola di approvazione ()AWS CLI

È possibile utilizzare il AWS CLI per associare un modello di regola di approvazione a uno o più repository.

Per associare un modello a un singolo repository

1. Dal terminale o dalla riga di comando, eseguire il comando `associate-approval-rule-template-with-repository`, specificando:
 - Il nome del modello di regola di approvazione che si desidera associare a un repository.
 - Il nome del repository da associare al modello di regola di approvazione.

Ad esempio, per associare un modello di regola di approvazione denominato **2-approver-rule-for-main** a un archivio denominato: **MyDemoRepo**

```
aws codecommit associate-approval-rule-template-with-repository --repository-name MyDemoRepo --approval-rule-template-name 2-approver-rule-for-main
```

2. In caso di successo, questo comando non restituisce alcun risultato.

Per associare un modello a più repository

1. Dal terminale o dalla riga di comando, eseguire il comando `batch-associate-approval-rule-template-with-repositories`, specificando:
 - Il nome del modello di regola di approvazione che si desidera associare a un repository.
 - I nomi dei repository da associare al modello di regola di approvazione.

Ad esempio, per associare un modello di regola di approvazione denominato **2-approver-rule-for-main** a un repository denominato **MyDemoRepo** e **MyOtherDemoRepo**:

```
aws codecommit batch-associate-approval-rule-template-with-repositories --repository-names "MyDemoRepo", "MyOtherDemoRepo" --approval-rule-template-name 2-approver-rule-for-main
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "associatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Gestire i modelli di regola di approvazione

È possibile gestire i modelli di regole di approvazione in un file Regione AWS per comprendere come vengono utilizzati e a cosa servono. Ad esempio, è possibile modificare i nomi e le descrizioni dei modelli di regola di approvazione per aiutare gli altri a capire il loro scopo. È possibile elencare tutti i modelli di regole di approvazione in un Regione AWS modello e ottenere informazioni sul contenuto e sulla struttura di un modello. È possibile esaminare quali modelli sono associati a un repository e quali repository sono associati a un modello.

Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, consulta [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#).

Gestisci i modelli di regole di approvazione (console)

Puoi visualizzare e gestire i modelli di regole di approvazione nella CodeCommit console.

Per gestire i modelli di regola di approvazione

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Scegli i modelli di regole di approvazione per visualizzare l'elenco dei modelli di regole di approvazione nel luogo in Regione AWS cui hai effettuato l'accesso.

Note

I modelli di regole di approvazione sono disponibili solo nel Regione AWS luogo in cui sono stati creati.

3. Se si desidera apportare modifiche a un modello, sceglierlo dall'elenco e quindi selezionare Edit (Modifica).

4. Apportare le modifiche, quindi fare clic su Save (Salva).

Gestisci i modelli di regole di approvazione (AWS CLI)

È possibile gestire i modelli di regola di approvazione con i seguenti comandi AWS CLI :

- [list-approval-rule-templates](#), per visualizzare un elenco di tutti i modelli di regole di approvazione in un Regione AWS
- [get-approval-rule-template](#), per visualizzare il contenuto di un modello di regola di approvazione
- [update-approval-rule-template-content](#), per modificare il contenuto di un modello di regola di approvazione
- [update-approval-rule-template-name](#), per modificare il nome di un modello di regola di approvazione
- [update-approval-rule-template-description](#), per modificare la descrizione di un modello di regola di approvazione
- [list-repositories-for-approval-rule-template](#), per visualizzare tutti i repository associati a un modello di regola di approvazione
- [list-associated-approval-rule-templates-for-repository](#), per visualizzare tutti i modelli di regola di approvazione associati a un repository

Per elencare tutti i modelli di regole di approvazione in un Regione AWS

1. Dal terminale o dalla riga di comando, esegui il comando `list-approval-rule-templates`. Ad esempio, per elencare tutti i modelli di regole di approvazione nella regione Stati Uniti orientali (Ohio):

```
aws codecommit list-approval-rule-templates --region us-east-2
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Per ottenere il contenuto di un modello di regola di approvazione

1. Nel terminale o nella riga di comando, eseguire il comando `get-approval-rule-template` specificando il nome del modello di regola di approvazione:

```
aws codecommit get-approval-rule-template --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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.\"
  }
}
```

Per aggiornare il contenuto di un modello di regola di approvazione

1. Al terminale o al prompt dei comandi, eseguire il comando `update-approval-rule-template-content`, specificando il nome del modello e il contenuto modificato. Ad esempio, per modificare il contenuto di un modello di regola di approvazione denominato **1-approver-rule** per ridefinire il pool di autorizzazioni agli utenti che assumono il ruolo di **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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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"
  }
}
```

Per aggiornare il nome di un modello di regola di approvazione

1. Al terminale o al prompt dei comandi, eseguire il comando `update-approval-rule-template-name`, specificando il nome corrente e il nome in cui si desidera modificarlo. Ad esempio, per modificare il nome di un modello di regola di approvazione **1-approver-rule** da **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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements
\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers
\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
  }
}
```

```

    "approvalRuleTemplateDescription": "All pull requests must be approved by
    one developer on the team.",
    "ruleContentSha256": "2f6c21a5cEXAMPLE"
  }
}

```

Per aggiornare la descrizione di un modello di regola di approvazione

1. Nel terminale o nella riga di comando, eseguire il comando `update-approval-rule-template-description`, specificando il nome del modello di regola di approvazione e la nuova descrizione:

```

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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```

{
  "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"
  }
}

```

Come elencare tutti i repository associati a un modello

1. Alla riga di comando o al terminale, eseguire il comando `list-repositories-for-approval-rule-template`, specificando il nome del modello:

```

aws codecommit list-repositories-for-approval-rule-template --approval-rule-
template-name 2-approver-rule-for-main

```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

Per elencare tutti i modelli associati a un repository

1. Alla riga di comando o al terminale, eseguire il comando `list-associated-approval-rule-templates-for-repository`, specificando il nome del repository:

```
aws codecommit list-associated-approval-rule-templates-for-repository --repository-name MyDemoRepo
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Dissociare un modello di regola di approvazione

Se le regole di approvazione generate da un modello di regola di approvazione non hanno più senso per il flusso di lavoro del team in un repository, è possibile disassociare il modello da tale repository. La disassociazione di un modello non rimuove le regole di approvazione create mentre il modello è stato associato al repository.

Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, consulta [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#)

Annulla l'associazione di un modello di regola di approvazione (console)

È possibile utilizzare la console per disassociare un repository da un modello di regola di approvazione.

Per disassociare un modello di regola di approvazione dai repository

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Scegliere i Approval rule templates (Modelli di regola di approvazione). Scegliere il modello che si desidera disassociare da un repository o da più repository, quindi scegliere Edit (Modifica).
3. In Associated Repositories (Repository associati), selezionare la X accanto ai repository che si desidera disassociare. I nomi del repository non vengono più visualizzati.
4. Seleziona Salva. Le regole di approvazione non vengono applicate alle richieste pull create in tali repository. Le regole sono ancora applicate per le richieste pull effettuate mentre l'associazione era in vigore.

Dissocia un modello di regola di approvazione ()AWS CLI

È possibile utilizzare il AWS CLI per dissociare uno o più repository da un modello di regola di approvazione.

Per disassociare un modello di regola di approvazione da un repository

1. Dal terminale o dalla riga di comando, eseguire il comando `disassociate-approval-rule-template-from-repository`, specificando:
 - Nome del modello di regola di approvazione.
 - Il nome del repository.

Ad esempio, per disassociare un modello di regola di approvazione denominato **1-approver-rule-for-all-pull-requests** da un repository denominato **MyDemoRepo**:

```
aws codecommit disassociate-approval-rule-template-from-repository --repository-name MyDemoRepo --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. In caso di successo, questo comando non restituisce alcun risultato.

Per disassociare un modello di regola di approvazione da più repository

1. Dal terminale o dalla riga di comando, eseguire il comando `batch-disassociate-approval-rule-template-from-repositories`, specificando:
 - Nome del modello di regola di approvazione.
 - I nomi dei repository.

Ad esempio, per disassociare un modello di regola di approvazione denominato **1-approver-rule-for-all-pull-requests** da un repository denominato **MyDemoRepo** e da un repository denominato **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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "disassociatedRepositoryNames": [  
    "MyDemoRepo",  
    "MyOtherDemoRepo"  
  ],  
  "errors": []  
}
```

Eliminare un modello di regola di approvazione

È possibile eliminare un modello di regola di approvazione se non lo si utilizza in alcun repository. L'eliminazione di modelli di regola di approvazione non utilizzati consente di mantenere organizzati i modelli e semplificare la ricerca di modelli che abbiano senso per i flussi di lavoro.

Per ulteriori informazioni sulle politiche gestite e sulle autorizzazioni per i modelli di regole di approvazione, consulta [Autorizzazioni per le azioni sui modelli di regole di approvazione](#) e [AWS politiche gestite per CodeCommit](#).

Argomenti

- [Eliminare un modello di regola di approvazione \(console\)](#)

- [Elimina un modello di regola di approvazione \(AWS CLI\)](#)

Eliminare un modello di regola di approvazione (console)

È possibile eliminare un modello di regola di approvazione se non è più rilevante per il lavoro di sviluppo. Quando si utilizza la console per eliminare un modello di regola di approvazione, questo viene disassociato da tutti i repository durante il processo di eliminazione.

Per eliminare un modello di regola di approvazione

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Scegliere i Approval rule templates (Modelli di regola di approvazione). Scegli il modello che desideri eliminare, quindi scegli Elimina.

Elimina un modello di regola di approvazione (AWS CLI)

È possibile utilizzare il AWS CLI per eliminare una regola di approvazione se è stata dissociata da tutti gli archivi. Per ulteriori informazioni, consulta [Dissocia un modello di regola di approvazione \(AWS CLI\)](#).

Per eliminare un modello di regola di approvazione

1. In un terminale o riga di comando, eseguire il comando `delete-approval-rule-template`, specificando il nome del modello di regola di approvazione che si desidera eliminare:

```
aws codecommit delete-approval-rule-template --approval-rule-template-name 1-approver-for-all-pull-requests
```

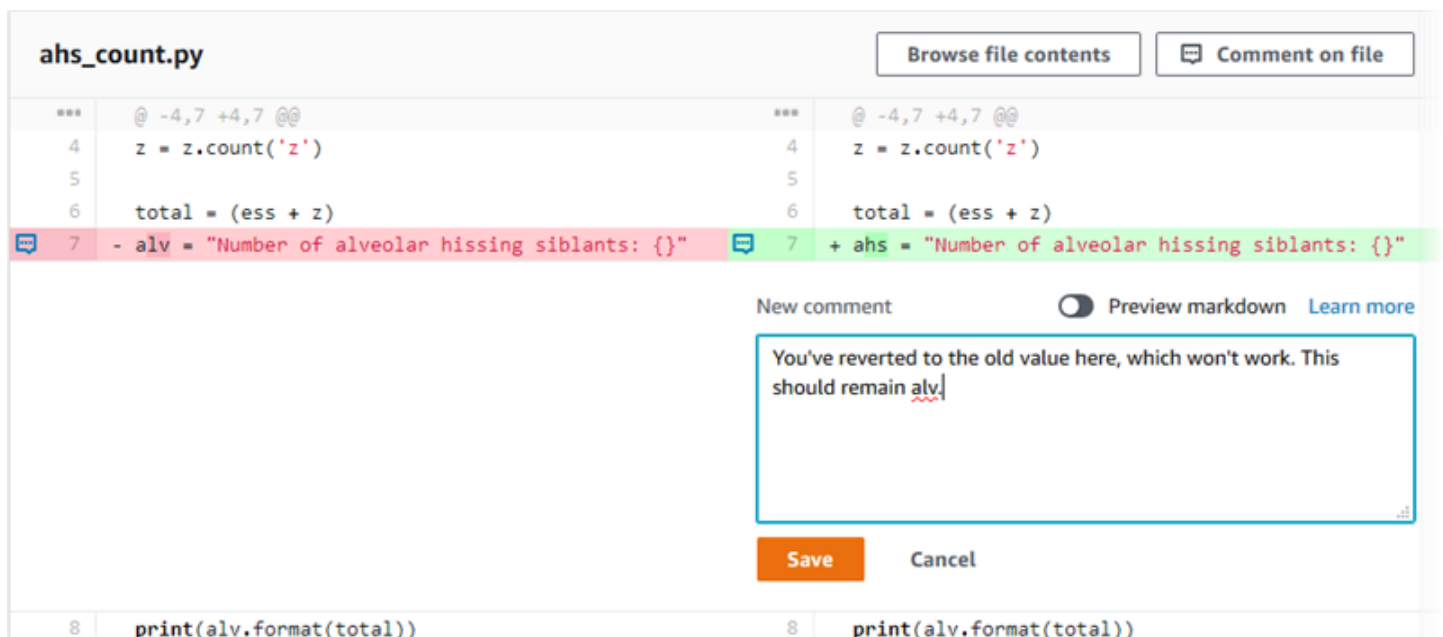
2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente. Se il modello di regola di approvazione è già stato eliminato, questo comando non restituisce nulla.

```
{
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

Lavorare con i commit nei repository AWS CodeCommit

I commit sono snapshot dei contenuti e modifiche apportate ai contenuti di un repository. Ogni volta che un utente esegue un commit e invia una modifica, l'informazione viene salvata e archiviata. Così anche le informazioni relative a chi ha eseguito il commit della modifica, la data e l'ora del commit e le modifiche apportate nell'ambito del commit. Puoi anche aggiungere tag ai commit per identificarli facilmente. In CodeCommit, puoi:

- Esaminare i commit.
- Visualizzare in un grafico la cronologia dei commit.
- Confrontare un commit con il padre o con un altro indicatore.
- Aggiungere commenti ai commit e rispondere ai commenti altrui.



Prima di inviare i commit a un CodeCommit repository, è necessario configurare il computer locale per la connessione al repository. Per il metodo più semplice, consulta [Per utenti HTTPS che utilizzano credenziali Git](#).

Per informazioni sull'utilizzo di altri aspetti del repository in CodeCommit, consulta,, [Lavorare con i repository](#)[Lavorare con i file](#), [Utilizzo delle richieste pull](#) e. [Gestione dei rami](#) [Utilizzo delle preferenze dell'utente](#)

Argomenti

- [Crea un commit in AWS CodeCommit](#)
- [Visualizza i dettagli del commit in AWS CodeCommit](#)
- [Confronta i commit in AWS CodeCommit](#)
- [Commenta un commit in AWS CodeCommit](#)
- [Crea un tag Git in AWS CodeCommit](#)
- [Visualizza i dettagli del tag Git in AWS CodeCommit](#)
- [Eliminare un tag Git in AWS CodeCommit](#)

Crea un commit in AWS CodeCommit

Quando si crea il primo commit per un nuovo repository, si utilizza il comando AWS CLI `put-file`. Questo crea il primo commit e consente di creare e specificare il ramo predefinito per il nuovo repository. Puoi usare Git o the AWS CLI per creare un commit in un CodeCommit repository. Se il repository locale è connesso a un CodeCommit repository, usi Git per inviare il commit dal repository locale al repository. CodeCommit Per creare un commit direttamente nella CodeCommit console, vedi e. [Creare o aggiungere un file a un AWS CodeCommit repository](#) [Modifica il contenuto di un file in un AWS CodeCommit repository](#)

Note

Come best practice, ti consigliamo di utilizzare le ultime versioni supportate di AWS CLI, Git e altri software. Se utilizzi il AWS CLI, assicurati che sia installata una versione recente per assicurarti di utilizzare una versione che contenga il `create-commit` comando.

Argomenti

- [Crea il primo commit per un repository utilizzando il AWS CLI](#)
- [Crea un commit usando un client Git](#)
- [Crea un commit usando il AWS CLI](#)

Crea il primo commit per un repository utilizzando il AWS CLI

È possibile utilizzare il AWS CLI `put-file` comando and per creare il primo commit per un repository. Using `put-file` crea un primo commit che aggiunge un file al repository vuoto e crea un ramo con il nome specificato. Designa il nuovo ramo come ramo predefinito per il repository.

 Note

Per utilizzare AWS CLI i comandi con CodeCommit, installa il. AWS CLI Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per creare il primo commit per un repository utilizzando il AWS CLI

1. Sul tuo computer locale, crea il file che desideri aggiungere come primo file al CodeCommit repository. Una pratica comune consiste nel creare un file README .md markdown che spieghi lo scopo di questo repository agli altri utenti del repository. Se includi un README .md file, il contenuto del file viene visualizzato automaticamente nella parte inferiore della pagina dei codici del repository nella console. CodeCommit
2. Dal terminale o dalla riga di comando, eseguire il comando put-file, specificando:
 - Il nome del repository in cui desideri aggiungere il primo file.
 - Il nome del ramo che desideri creare come ramo predefinito.
 - Il percorso locale del file. La sintassi utilizzata per questo percorso varia a seconda del sistema operativo locale.
 - Il nome del file da aggiungere, incluso il percorso in cui il file aggiornato è archiviato nel repository.
 - Il nome utente e l'e-mail che desideri associare a questo file.
 - Un messaggio di commit che spiega perché è stato aggiunto il file.

Il nome utente, l'indirizzo e-mail e il messaggio di commit sono facoltativi, ma possono aiutare gli altri utenti a sapere chi ha apportato la modifica e perché. Se non fornisci un nome utente, per CodeCommit impostazione predefinita utilizza il tuo nome utente IAM o una derivazione del login della console come nome dell'autore.

Ad esempio, per aggiungere un file denominato *README.md* con il contenuto del file codificato in base 6 di esempio a un repository denominato un ramo denominato *MyDemoRepo: 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."
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "commitId": "724caa36EXAMPLE",
  "blobId": "a8a94062EXAMPLE",
  "treeId": "08b2fc73EXAMPLE"
}
```

Crea un commit usando un client Git

Puoi creare commit usando un client Git installato sul tuo computer locale e poi inviarli al tuo CodeCommit repository.

1. Completare i prerequisiti, tra cui [Configurazione](#).

Important

Se non è stata completata la configurazione, non è possibile collegarsi o eseguire il commit nel repository usando Git.

2. Assicurarsi che un commit sia stato creato nel ramo corretto. Per consultare un elenco di rami disponibili e scoprire quale ramo è attualmente impostato per l'utilizzo, eseguire `git branch`. Vengono visualizzati tutti i rami. Viene visualizzato un asterisco (*) accanto al ramo corrente. Per passare a un altro ramo, eseguire `git checkout branch-name`. Se questo è il tuo primo commit, esegui il `git config` comando per configurare il tuo client Git per creare un ramo iniziale con il nome che vuoi usare per quel ramo. Ad esempio, se vuoi che il tuo ramo predefinito abbia il nome **development**:

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

Tip

Questo comando è disponibile solo in Git v.2.28 e versioni successive.

Puoi anche eseguire questo comando per impostare il nome del ramo predefinito su tutti **development** i repository appena creati:

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

3. Apportare una modifica al ramo (ad esempio aggiungere, modificare o eliminare un file).

Ad esempio, nel repository locale, create un file denominato `bird.txt` con il seguente testo:

```
bird.txt
-----
Birds (class Aves or clade Avialae) are feathered, winged, two-legged, warm-blooded, egg-laying vertebrates.
```

4. Eseguire `git status`, che indica che `bird.txt` non è ancora stato incluso in alcun commit in sospeso:

```
...
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    bird.txt
```

5. Eseguire `git add bird.txt` per includere il nuovo file nel commit in sospeso.
6. Se si esegue di nuovo `git status`, viene visualizzato un output simile al seguente. Indica che `bird.txt` fa ora parte del commit in sospeso o è nella gestione temporanea del commit:

```
...
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   bird.txt
```

7. Per finalizzare il commit, eseguire `git commit` con l'opzione `-m` (ad esempio `git commit -m "Adding bird.txt to the repository."`). L'opzione `-m` crea il messaggio di commit.
8. Se si esegue di nuovo `git status`, viene visualizzato un output simile al seguente. Indica che il commit è pronto per essere inviato dal repository locale al repository: CodeCommit

```
...
nothing to commit, working directory clean
```

- Prima di inviare il commit finalizzato dal repository locale al CodeCommit repository, puoi vedere cosa stai inviando eseguendo `git diff --stat remote-name/branch-name, remote-name` dov'è il soprannome che il repository locale usa per il CodeCommit repository e **branch-name** il nome del ramo da confrontare.

Tip

Per ottenere il nome alternativo, eseguire `git remote`. Per ottenere un elenco di nomi di rami, eseguire `git branch`. Viene visualizzato un asterisco (*) accanto al ramo corrente. È inoltre possibile eseguire `git status` per ottenere il nome del ramo corrente.

Note

Se hai clonato il repository, dal punto di vista del repository locale, non è il nome del repository. **remote-name** CodeCommit Quando si clona un repository, viene impostato automaticamente su. **remote-name** origin

Ad esempio, `git diff --stat origin/main` restituisce un output simile al seguente:

```
bird.txt | 1 +
1 file changed, 1 insertion(+)
```

L'output presuppone che tu abbia già collegato il repository locale al repository. CodeCommit Per istruzioni, consultare [Connessione a un repository](#).

- Quando sei pronto per inviare il commit dal repository locale al CodeCommit repository, esegui `git push remote-name branch-name, remote-name` dov'è il soprannome che il repository locale usa per il CodeCommit repository ed **branch-name** è il nome del ramo da inviare al repository. CodeCommit

Ad esempio, l'esecuzione di `git push origin main` restituisce un output simile al seguente:

Per 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
```

Per 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

Se si aggiunge l'opzione `-u` a `git push` (ad esempio `git push -u origin main`), è necessario solo eseguire `git push` in futuro per impostare le informazioni di registrazione upstream. Per ottenere le informazioni di registrazione upstream, eseguire `git remote show remote-name` (ad esempio, `git remote show origin`).

Per altre opzioni, consulta la documentazione di Git.

Crea un commit usando il AWS CLI

È possibile utilizzare il AWS CLI `create-commit` comando and per creare un commit per un repository sulla punta di un ramo specificato. Puoi anche creare un commit di unione senza riferimenti per rappresentare i risultati dell'unione di due specificatori di commit. Per ulteriori informazioni, consulta [Creazione di un commit senza riferimenti](#).

Note

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Come creare un commit

1. Nel computer locale, apportare le modifiche che devono essere sottoposte a commit nel repository CodeCommit.
2. Dal terminale o dalla riga di comando, eseguire il comando `create-commit`, specificando:
 - Il repository in cui si desidera eseguire il commit delle modifiche.
 - Il ramo in cui si desidera eseguire il commit delle modifiche.
 - L'ID commit completo dell'ultimo commit effettuato al ramo, noto anche come commit di estremità o testa o l'ID commit dell'elemento padre.
 - Se mantenere cartelle vuote se le modifiche apportate eliminano il contenuto di queste cartelle. Questo valore è `false` per impostazione predefinita.
 - Le informazioni sui file che si desidera aggiungere, modificare o eliminare.
 - Il nome utente e l'indirizzo e-mail che si desidera associare a queste modifiche.
 - Un messaggio di commit che illustra perché sono state apportate queste modifiche.

Il nome utente, l'indirizzo e-mail e il messaggio di commit sono facoltativi, ma consentono agli altri utenti di sapere chi ha apportato le modifiche e il motivo. Se non fornisci un nome utente, per CodeCommit impostazione predefinita utilizza il tuo nome utente IAM o una derivazione dell'accesso alla console come nome dell'autore.

Ad esempio, per creare un commit per un repository che aggiunge un `README.md` file a un repository denominato nel ramo `MyDemoRepo main` Il contenuto del file è in Base64 e recita «Benvenuto nel repository del nostro team!» :

```
aws codecommit create-commit --repository-name MyDemoRepo --  
branch-name main --parent-commit-id 4c925148EXAMPLE --put-files  
"filePath=README.md,fileContent=V2VsY29tZSB0byBvdXIgdGVhbSBvZXBvc2l0b3J5IQo="
```

Tip

Per ottenere l'ID di commit principale, esegui il comando [get-branch](#).

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "commitId": "4df8b524-EXAMPLE",
  "treeId": "55b57003-EXAMPLE",
  "filesAdded": [
    {
      "blobId": "5e1c309dEXAMPLE",
      "absolutePath": "meeting.md",
      "fileMode": "NORMAL"
    }
  ],
  "filesDeleted": [],
  "filesUpdated": []
}
```

Per creare un commit che apporti modifiche ai file denominati *file1.py* and *file2.txt*, rinomini un file da *picture.png* a *image1.png* e lo sposti da una directory denominata *pictures* a una directory denominata *images*, ed elimini un file denominato *ExampleSolution.py* in un repository denominato *MyDemoRepo* su un ramo denominato il *MyFeatureBranch* cui commit più recente ha un ID di: *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 sintassi del `--put-files` segmento varia a seconda del sistema operativo. L'esempio precedente è ottimizzato per utenti Linux, macOS o Unix e utenti Windows con un emulatore Bash. Gli utenti di Windows sulla riga di comando o in Powershell devono utilizzare una sintassi appropriata per tali sistemi.

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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"
    }
  ]
}
```

Visualizza i dettagli del commit in AWS CodeCommit

È possibile utilizzare la AWS CodeCommit console per sfogliare la cronologia dei commit in un repository. Questo può aiutarti a individuare le modifiche apportate in un repository, tra cui:

- Quando sono state apportate le modifiche e chi le ha apportate.
- Quando i commit specifici sono stati uniti in un ramo.

Visualizzazione della cronologia di commit per un ramo potrebbe anche aiutarti a comprendere la differenza tra i rami. Se utilizzi il tagging, puoi anche visualizzare rapidamente il commit che è stato contrassegnato da un tag e da elementi padre di quel commit con tag. Nella riga di comando, puoi usare Git per visualizzare i dettagli sui commit in un repository locale o in un repository. CodeCommit

Sfoglia i commit in un repository

È possibile utilizzare la AWS CodeCommit console per sfogliare la cronologia dei commit in un repository. È anche possibile visualizzare un grafico dei commit nel repository e i suoi rami nel corso del tempo. Questo può aiutarti a comprendere la storia del repository, ad esempio quando sono state apportate le modifiche.

Note

L'utilizzo del comando `git rebase` per un rebase delle modifiche di un repository potrebbe causare il malfunzionamento dei commit. Per ulteriori informazioni, consulta [Rebase del ramo Git](#) o la documentazione Git.

Argomenti

- [Sfoglia la cronologia dei commit di un repository](#)
- [Visualizza un grafico della cronologia dei commit di un repository](#)

Sfoglia la cronologia dei commit di un repository

È possibile cercare la cronologia di commit per un determinato ramo o tag del repository, incluse informazioni sul committer e sul messaggio di commit. È anche possibile visualizzare il codice per un commit.

Per cercare la cronologia dei commit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository per il quale si desidera esaminare la cronologia di commit.
3. Nel riquadro di navigazione selezionare Commits (Commit). Nella visualizzazione della cronologia di commit viene visualizzata la cronologia dei commit per il repository del ramo predefinito, nell'ordine cronologico inverso rispetto alla data di commit. La data e l'ora sono in formato UTC. È possibile visualizzare la cronologia dei commit di un ramo differente, selezionando il pulsante di selezione delle visualizzazioni, quindi scegliendo un ramo dall'elenco. Se stai utilizzando i tag nel repository, puoi visualizzare un commit con un tag specifico e i relativi elementi padre scegliendo quel tag nel pulsante di selezione della visualizzazione.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Commits

MyDemoRepo

master

Commits

Commit visualizer

Compare commits

Commits

< 1 ... >

Commit ID	Commit message	Commit date	Author	Actions
d615e7ae	Merge branch 'AnotherBranch' into testbranch	5 days ago	Maria Garcia	Copy ID Browse
b6589863	Added another file.	5 days ago	Li Juan	Copy ID Browse
73a6e39c	Merge branch 'master' into testbranch	5 days ago	Maria Garcia	Copy ID Browse
6bbb6d3c	Another test of the editing feature.	5 days ago	Li Juan	Copy ID Browse
edacdffe	Testing this out to see how well it works.	5 days ago	Li Juan	Copy ID Browse
70bb94d7	Revised test results with correct information.	5 days ago	Li Juan	Copy ID Browse
b78e6d1c	Merge branch 'master' into testbranch	5 days ago	Maria Garcia	Copy ID Browse
84b7d158	Edited ahs_count.py	22 days ago	Maria Garcia	Copy ID Browse

4. Per visualizzare la differenza tra un commit e il relativo padre e per visualizzare i commenti sulle modifiche, scegli l'ID di commit abbreviato. Per ulteriori informazioni, consultare [Confronta un commit con il suo genitore](#) e [Commentare un commit](#). Per visualizzare la differenza tra un commit e qualsiasi altro specificatore di commit, tra cui un ramo, un tag o un ID di commit, consulta [Confrontare due specificatori di commit](#).
5. Effettuare una o più delle seguenti operazioni:
 - Per visualizzare la data e l'ora di una modifica, passa con il mouse sulla data di commit.

- Per visualizzare l'intero ID di commit, copialo, quindi incollalo in un editor di testo o in altre posizioni. Per copiarlo, scegli Copia ID.
- Per visualizzare il codice così com'era al momento del commit, selezionare Browse (Cerca). Il contenuto del repository al momento del commit viene visualizzato nella visualizzazione Code (Codice). Il selettore di visualizzazione mostra il pulsante ID di commit abbreviato invece di un ramo o tag.

Visualizza un grafico della cronologia dei commit di un repository

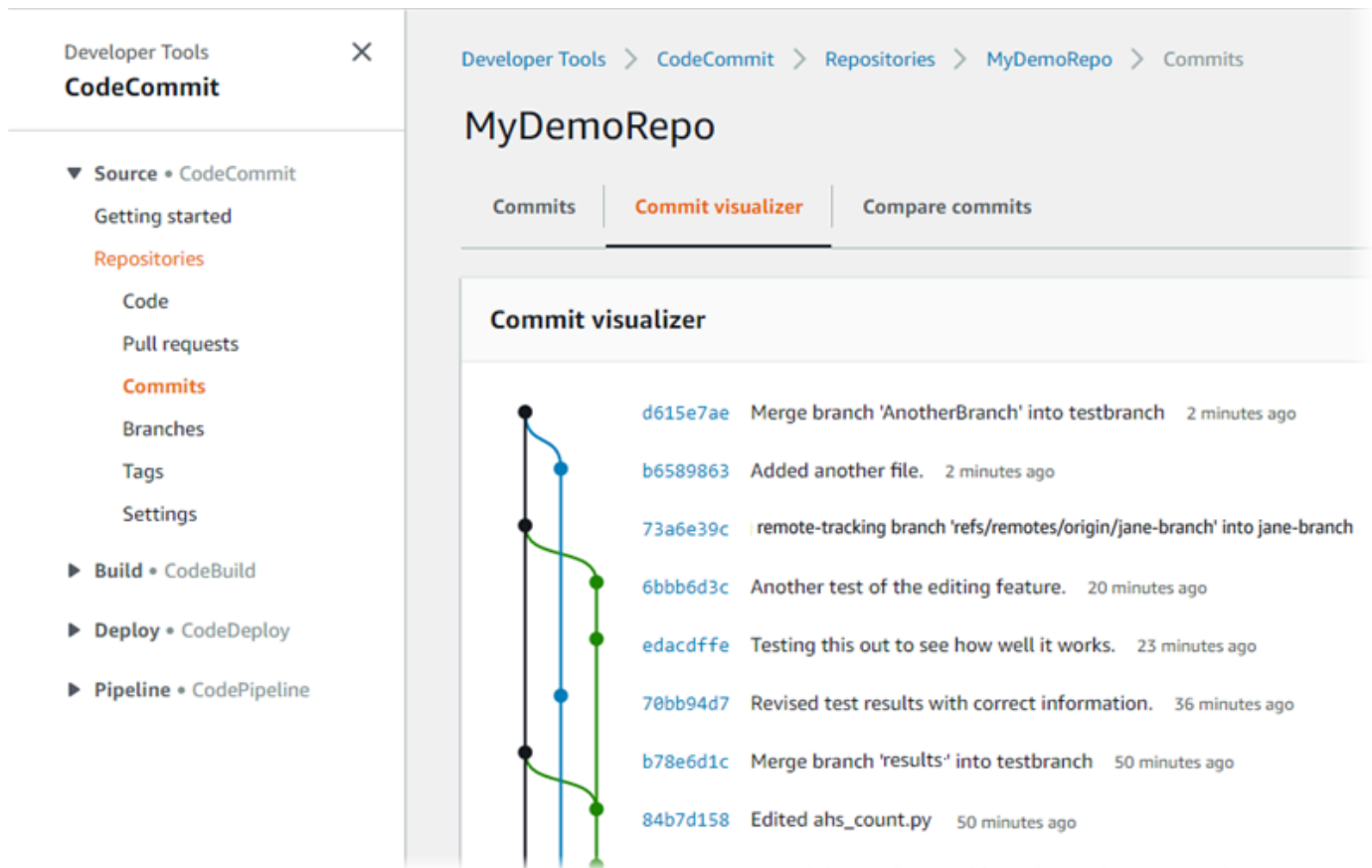
Puoi visualizzare un grafico dei commit eseguiti su un repository. La vista Commit Visualizer (Visualizzatore commit) è la rappresentazione di un grafo aciclico orientato (DAG) di tutti i commit eseguiti su un ramo del repository. Questa rappresentazione grafica può aiutarti a comprendere quando viene eseguito il commit e le caratteristiche associate che sono state aggiunte o unite. Può aiutarti anche a comprendere quando viene eseguita una modifica in relazione alle altre modifiche.

Note

I commit uniti utilizzando il metodo di avanzamento rapido non vengono visualizzati come righe separate nel grafico di commit.

Per visualizzare un grafico di commit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository per il quale si desidera visualizzare il grafico di commit.
3. Nel riquadro di navigazione, scegliere Commits (Commit), quindi selezionare la scheda Commit visualizer (Visualizzatore commit).



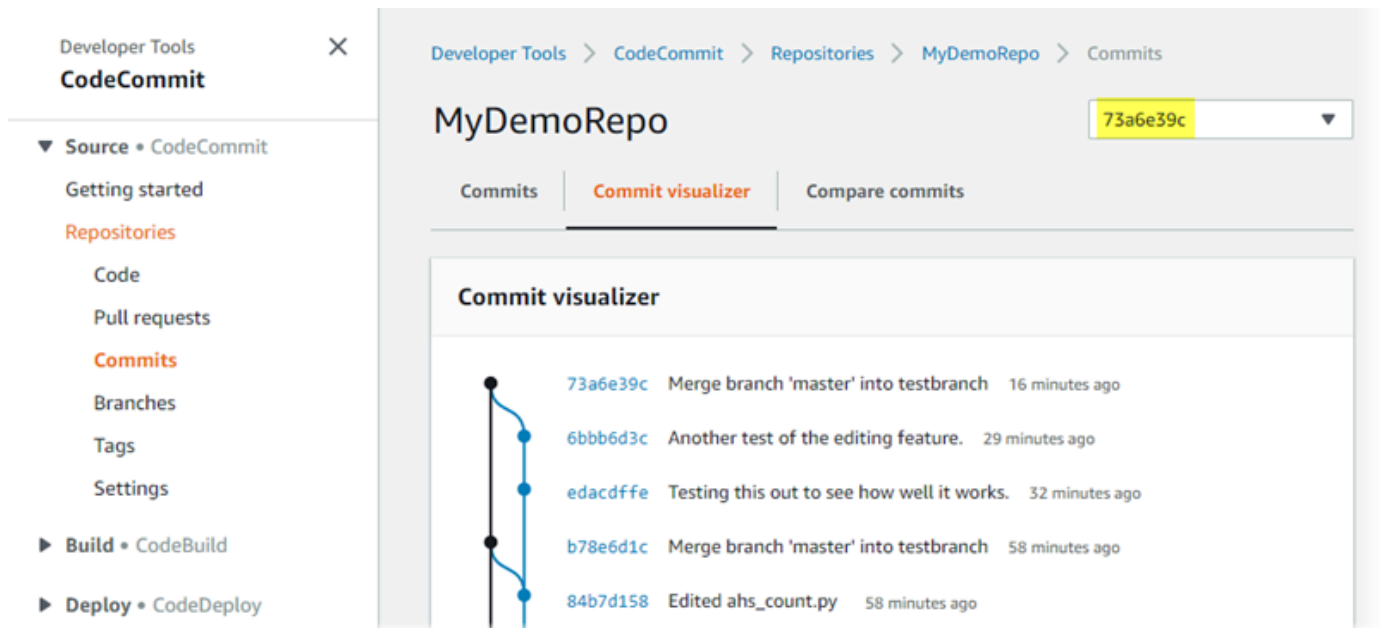
Nel grafico di commit, l'ID di commit abbreviato e l'oggetto di ogni messaggio di commit appaiono accanto a quel punto del grafico.

Note

Il grafico può mostrare fino a 35 rami su una pagina. Se sono presenti più di 35 rami, il grafico è troppo complesso per essere visualizzato. È possibile semplificare la visualizzazione in due modi:

- Utilizzando il pulsante dello strumento di selezione della visualizzazione per visualizzare il grafico per un determinato ramo.
- Incollando l'ID di commit completo nella casella di ricerca per eseguire il rendering del grafico da quel commit.

4. Per eseguire il rendering di un nuovo grafico da un commit, scegliere il punto del grafico che corrisponde al commit. Il pulsante di selezione della visualizzazione passa all'ID commit abbreviato.



Visualizza i dettagli del commit ()AWS CLI

Git consente di visualizzare i dettagli sui commit. È inoltre possibile utilizzare il AWS CLI per visualizzare i dettagli sui commit in un repository locale o in un CodeCommit repository eseguendo i seguenti comandi:

- Per visualizzare informazioni su un commit, eseguire [aws codecommit get-commit](#).
- Per visualizzare informazioni su più commit, eseguire [aws codecommit batch-get-commits](#).
- Per visualizzare le informazioni su un commit di unione, eseguire [aws codecommit get-merge-commit](#).
- Per visualizzare informazioni sulle modifiche relative a un identificatore di commit (branch, tag, HEAD o altri riferimenti completi, come commit IDs), esegui. [aws codecommit get-differences](#)
- Per visualizzare i contenuti con codifica base64 di un oggetto blob Git in un repository, eseguire [aws codecommit get-blob](#).

Per visualizzare le informazioni su un commit

1. Eseguire il comando `aws codecommit get-commit` specificando:
 - Il nome del CodeCommit repository (con l'opzione `--repository-name`).
 - L'ID commit completo.

Ad esempio, per visualizzare informazioni su un commit con l'ID 317f8570EXAMPLE in un CodeCommit repository denominato: MyDemoRepo

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id
317f8570EXAMPLE
```

2. In caso di esito positivo, l'output di questo comando include quanto segue:

- Le informazioni relative all'autore del commit (come configurato in Git), tra cui la data nel formato timestamp e l'offset UTC.
- Le informazioni sul committer (come configurato in Git), tra cui la data nel formato timestamp e l'UTC offset.
- L'ID della struttura Git in cui esiste il commit.
- L'ID commit del commit dell'elemento padre.
- Il messaggio commit.

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "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"
  }
}
```

Per visualizzare le informazioni su un commit di unione

1. Eseguire il comando `get-merge-commit` specificando:

- Uno specificatore di commit per l'origine dell'unione (con l'opzione `--source-commit-specifier`).
- Uno specificatore di commit per la destinazione dell'unione (con l'opzione `--destination-commit-specifier`).
- L'opzione di unione da utilizzare (con l'opzione `--merge-option`).
- Il nome del repository (con l'opzione `--repository-name`).

Ad esempio, per visualizzare le informazioni su un commit di unione per il ramo di origine denominato *bugfix-bug1234* con un ramo di destinazione denominato *main* utilizzando la *THREE_WAY_MERGE* strategia in un repository denominato: *MyDemoRepo*

```
aws codecommit get-merge-commit --source-commit-specifier bugfix-bug1234 --
destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-
name MyDemoRepo
```

2. Se il comando viene eseguito correttamente, l'output restituisce informazioni simili alle seguenti:

```
{
  "sourceCommitId": "c5709475EXAMPLE",
  "destinationCommitId": "317f8570EXAMPLE",
  "baseCommitId": "fb12a539EXAMPLE",
  "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

Come visualizzare informazioni su più commit

1. Eseguire il comando `batch-get-commits` specificando:

- Il nome del CodeCommit repository (con l'opzione `--repository-name`).
- Un elenco completo di commit IDs per ogni commit su cui si desidera visualizzare informazioni.

Ad esempio, per visualizzare informazioni sui commit con IDs *317f8570EXAMPLE* e *4c925148EXAMPLE* in un CodeCommit repository denominato: *MyDemoRepo*

```
aws codecommit batch-get-commits --repository-name MyDemoRepo --commit-ids
317f8570EXAMPLE 4c925148EXAMPLE
```

2. In caso di esito positivo, l'output di questo comando include quanto segue:

- Le informazioni relative all'autore dei commit (come configurato in Git), tra cui la data nel formato timestamp e l'offset UTC.
- Le informazioni sui committer (come configurato in Git), tra cui la data nel formato timestamp e l'UTC offset.
- L' IDs albero Git in cui esiste il commit.
- Il commit IDs del commit principale.
- I messaggi commit.

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "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"
}
}
```

Per visualizzare informazioni sulle modifiche per un specificatore di commit

1. Eseguire il comando `aws codecommit get-differences` specificando:
 - Il nome del CodeCommit repository (con l'opzione `--repository-name`).
 - Gli specificatori di commit sui quali si desidera ottenere informazioni. È richiesto solo `--after-commit-specifier`. Se non si specifica `--before-commit-specifier`, vengono visualizzati tutti i file correnti relativi a `--after-commit-specifier`.

Ad esempio, per visualizzare informazioni sulle differenze tra i commit con IDs `317f8570EXAMPLE` e `4c925148EXAMPLE` in un CodeCommit repository denominato: `MyDemoRepo`

```
aws codecommit get-differences --repository-name MyDemoRepo --before-commit-specifier 317f8570EXAMPLE --after-commit-specifier 4c925148EXAMPLE
```

2. In caso di esito positivo, l'output di questo comando include quanto segue:
 - Un elenco di differenze, tra cui il tipo di modifica (A per aggiunto, D per eliminato o M per modificato).
 - Le modalità del tipo di modifica del file.

- L'ID dell'oggetto blob Git che contiene la modifica.

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3bEXAMPLE",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf28fEXAMPLE",
        "mode": "100644"
      }
    }
  ]
}
```

Come visualizzare le informazioni su un oggetto blob Git

1. Eseguire il comando `aws codecommit get-blob` specificando:
 - Il nome del CodeCommit repository (con l'opzione `--repository-name`).
 - L'ID del blob Git (con l'opzione `--blob-id`).

Ad esempio, per visualizzare informazioni su un blob Git con l'ID di `2eb4af3bEXAMPLE` in un CodeCommit repository denominato: `MyDemoRepo`

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

2. In caso di esito positivo, l'output di questo comando include quanto segue:
 - Il contenuto con codifica base64 del blob, in genere un file.

Ad esempio, l'output del comando precedente dovrebbe essere simile all'esempio seguente:

```
{
  "content": "QSBcW5hcnkgTGZyToEXAMPLE="
}
```

Visualizza i dettagli del commit (Git)

Prima di seguire questi passaggi, dovresti aver già collegato il repository locale al CodeCommit repository e aver eseguito le modifiche. Per istruzioni, consulta [Connessione a un repository](#).

Per mostrare le modifiche apportate al commit più recente in un repository, esegui il comando. `git show`

```
git show
```

Il comando genera un output simile al seguente:

```
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 0000000..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

In questo e nei seguenti esempi, commit è IDs stato abbreviato. I commit completi non IDs vengono visualizzati.

Per visualizzare le modifiche effettuate, utilizzare il comando `git show` con l'ID di commit:

```
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.
```

Per vedere le differenze tra due commit, esegui il `git diff` comando e includi i due commit IDs.

```
git diff ce22850d 4f8c6f9d
```

In questo esempio, la differenza tra i due commit è data dal fatto che sono stati aggiunti due file. Il comando genera un output simile al seguente:

```
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
```


Per usare Git per visualizzare i dettagli sui commit in un repository locale, esegui il git log comando:

```
git log
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
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
```

Per mostrare solo i commit IDs e i messaggi, esegui il git log --pretty=oneline comando:

```
git log --pretty=oneline
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
94ba1e60 Added horse.txt
4c925148 Added cat.txt and dog.txt
```

Per altre opzioni, consulta la documentazione di Git.

Confronta i commit in AWS CodeCommit

È possibile utilizzare la CodeCommit console per visualizzare le differenze tra gli specificatori di commit in un repository. CodeCommit Puoi visualizzare rapidamente la differenza tra un commit e il relativo padre. Puoi anche confrontare due riferimenti qualsiasi, incluso il commit. IDs

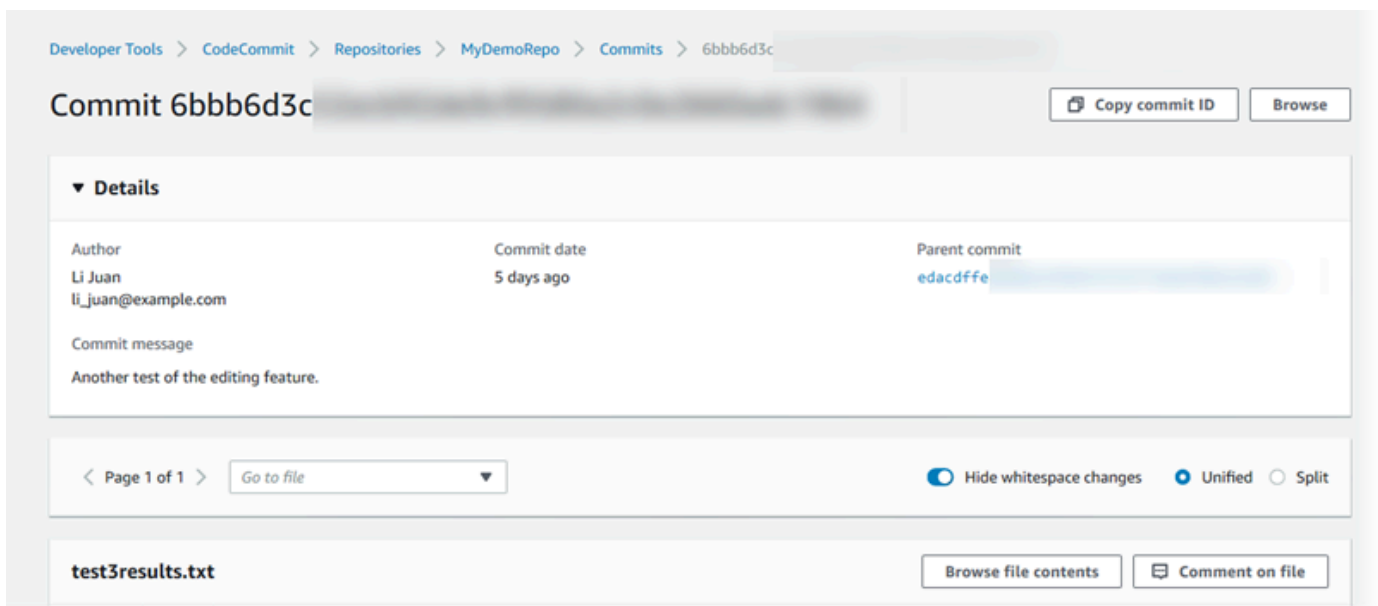
Argomenti

- [Confronta un commit con il suo genitore](#)
- [Confrontare due specificatori di commit](#)

Confronta un commit con il suo genitore

Puoi visualizzare rapidamente la differenza tra un commit e il relativo padre per esaminare il messaggio di commit, il committer e cosa è cambiato.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nella pagina Repositories (Repository), scegliere il repository in cui si desidera visualizzare la differenza tra un commit e il relativo padre.
3. Nel riquadro di navigazione selezionare Commits (Commit).
4. Scegliere l'ID commit abbreviato di qualsiasi commit nell'elenco. La visualizzazione cambia per mostrare i dettagli di questo commit, incluse le differenze tra il commit e il commit padre.



È possibile mostrare le modifiche nei file affiancati (visualizzazione Split (Doppia)) o inline (visualizzazione Unified (Unificata)). È anche possibile nascondere o visualizzare le modifiche degli spazi vuoti. È inoltre possibile aggiungere commenti. Per ulteriori informazioni, consulta [Commentare un commit](#).

Note

Le preferenze per la visualizzazione del codice e per altre impostazioni della console vengono salvate come cookie del browser a ogni modifica. Per ulteriori informazioni, consulta [Utilizzo delle preferenze dell'utente](#).

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Commits > 7d09e44c

Commit 7d09e44c

Copy commit ID

Browse

▼ Details

Author

Mary Major

mary_major@example.com

Commit date

48 minutes ago

Parent commit

[e6aca768](#)

Commit message

Adding a readme file to the repository.

< Page 1 of 1 > ▼

☒ Hide whitespace changes

☒ Unified ☐ Split

readme.md

Browse file contents

Comment on file

1

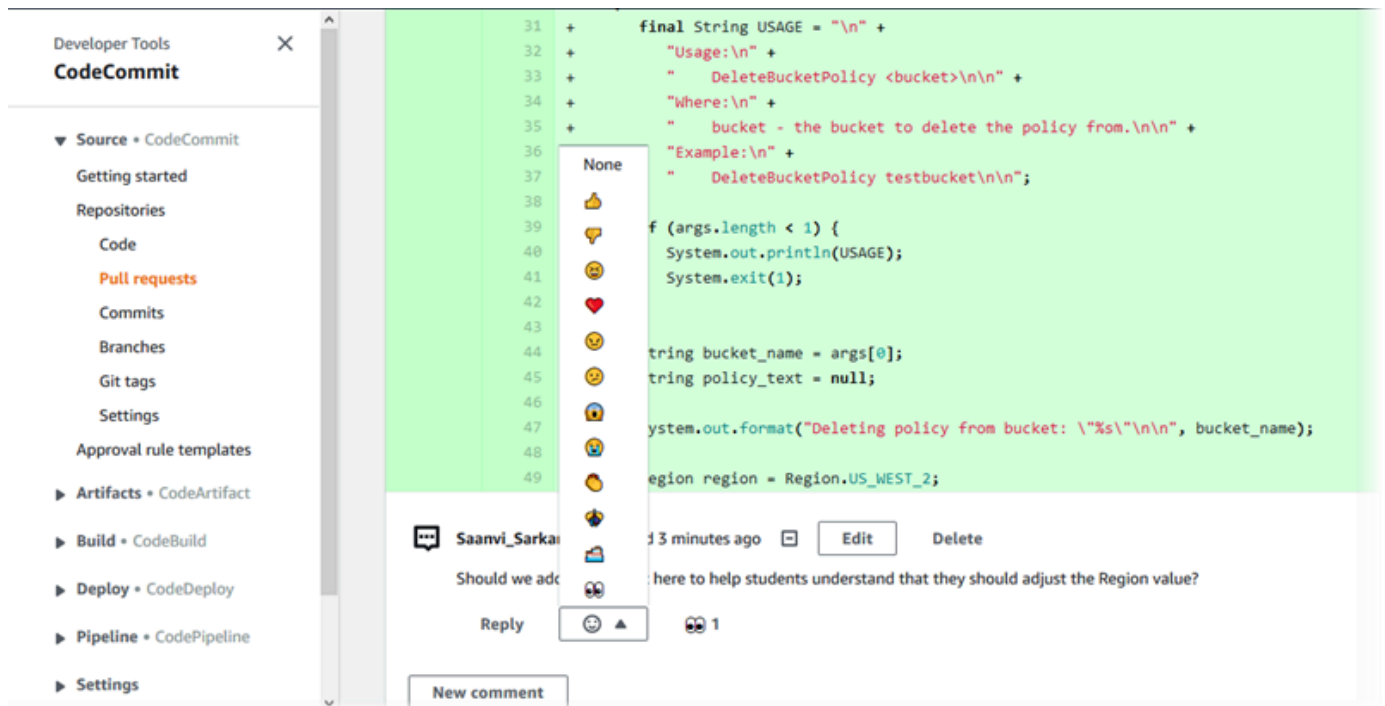
- This is a readme file that provides a basic description of what's in this repository.

\ No newline at end of file

1

+ Use this repository for code changes to the **Demo** project. The default branch is **master**. Cod

\ No newline at end of file



Note

A seconda dello stile finale della riga, dell'editor di codice e da altri fattori, è possibile che vengano visualizzate intere righe aggiunte o eliminate anziché modifiche specifiche di una riga. Il livello di dettaglio corrisponde a quello che viene restituito nel comando `git show` o `git diff`.

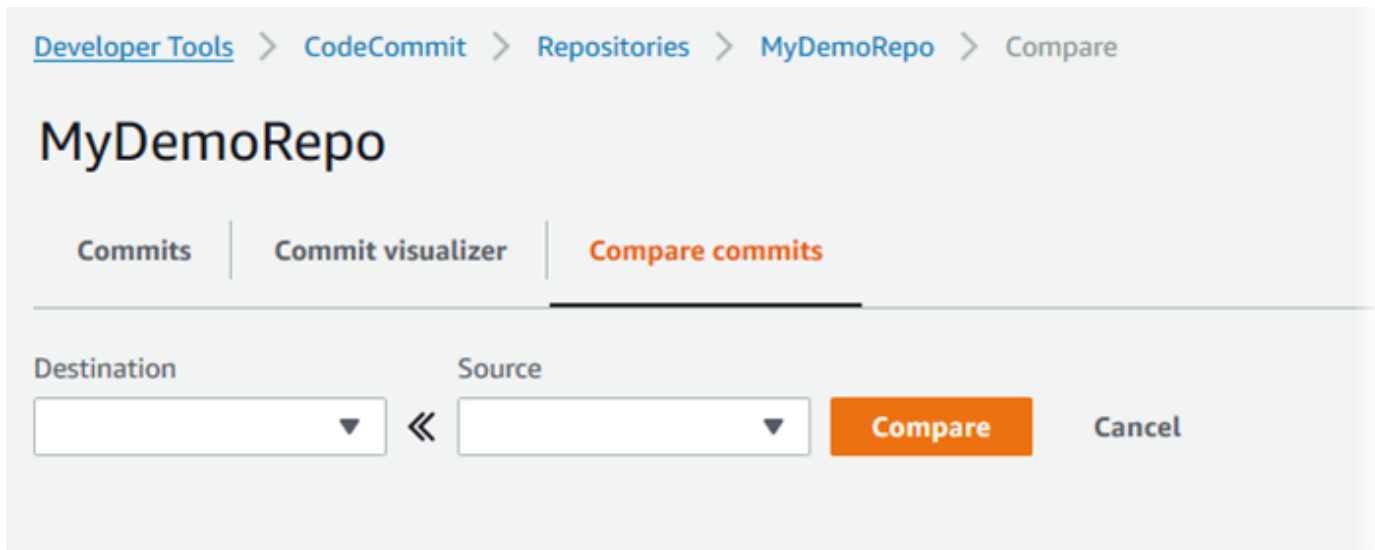
5. Per confrontare un commit con il relativo padre dalla scheda Commit visualizer (Visualizzatore commit), scegliere l'ID commit abbreviato. Verranno visualizzati i dettagli del commit, comprese le modifiche tra il commit e il relativo padre.

Confrontare due specificatori di commit

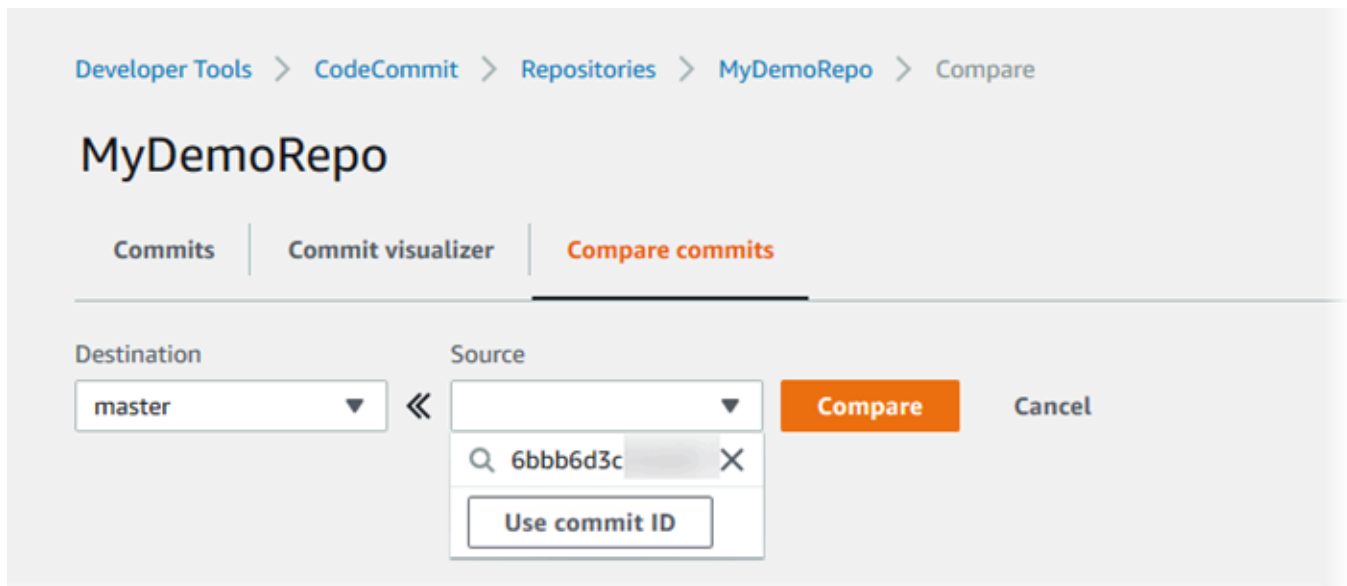
È possibile visualizzare le differenze tra due specificatori di commit qualsiasi nella CodeCommit console. Gli specificatori di commit sono riferimenti, come branch, tags e commit. IDs

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Nella pagina Repositories (Repository), selezionare il repository in cui si desidera confrontare commit, rami o commit con tag.

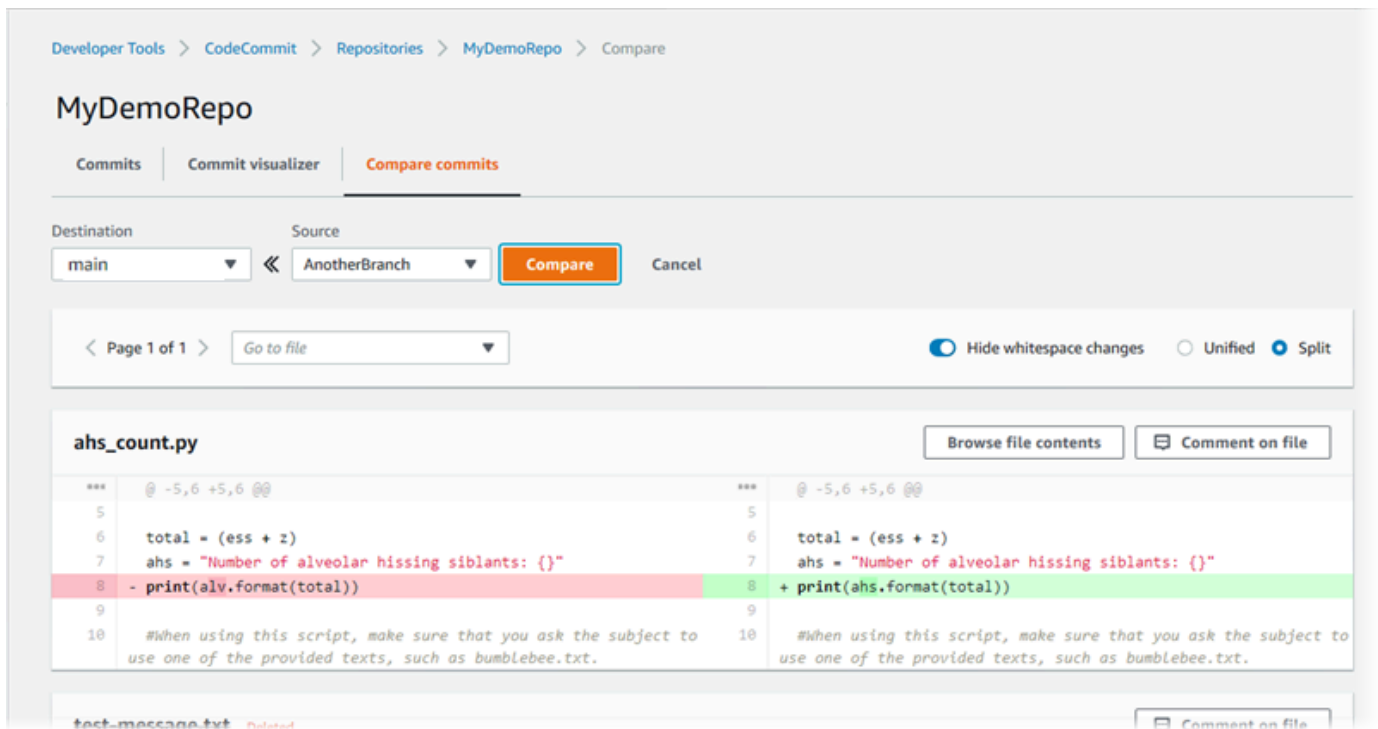
3. Nel riquadro di navigazione scegliere Commits (Commit), quindi selezionare Compare commits (Confronta commit).



4. Utilizza le caselle per confrontare due specificatori di commit.
- Per confrontare l'estremità di un ramo, scegli il nome del ramo dall'elenco. In tal modo viene selezionato il commit più recente dal ramo per il confronto.
 - Per confrontare un commit con un determinato tag associato, scegliere il nome tag dall'elenco. In tal modo viene selezionato il commit con tag per il confronto.
 - Per confrontare un determinato commit, immettere o incollare l'ID commit nella casella. Per ottenere l'ID commit completo, scegliere Commits (Commit) nella barra di navigazione e copiare l'ID commit dall'elenco. Nella pagina Confronta i commit, incolla l'ID di commit completo nella casella di testo e scegli Usa commit ID.



5. Dopo aver selezionato gli specificatori, scegliere Compare (Confronta).



È possibile mostrare le differenze nei file affiancati (visualizzazione Split (Doppia)) o inline (visualizzazione Unified (Unificata)). È anche possibile nascondere o visualizzare le modifiche degli spazi vuoti.

6. Per annullare le opzioni di confronto, scegliere Cancel (Annulla).

Commenta un commit in AWS CodeCommit

È possibile utilizzare la CodeCommit console per commentare i commit in un repository e visualizzare e rispondere ai commenti di altri utenti sui commit. Questo può aiutarti a discutere le modifiche apportate in un repository, tra cui:

- Perché sono state apportate modifiche.
- Se sono necessarie più modifiche.
- Se le modifiche devono essere unite in un altro ramo.

Puoi commentare un intero commit, un file in un commit oppure una riga o una modifica specifica di un file. È inoltre possibile creare un collegamento a una riga di codice selezionando la riga e quindi copiando l'URL risultante nel browser.

Note

Per ottenere i migliori risultati, usa la funzione di commento quando accedi come utente IAM. La funzionalità dei commenti non è ottimizzata per gli utenti che effettuano l'accesso con le credenziali dell'account radice, l'accesso federato o le credenziali provvisorie.

Argomenti

- [Visualizza i commenti su un commit in un repository](#)
- [Aggiungi e rispondi ai commenti su un commit in un repository](#)
- [Visualizza, aggiungi, aggiorna e rispondi ai commenti \(\)AWS CLI](#)

Visualizza i commenti su un commit in un repository

È possibile utilizzare la CodeCommit console per visualizzare i commenti su un commit.

Come visualizzare i commenti su un commit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository per il quale si desidera esaminare i commenti sui commit.

3. Nel riquadro di navigazione selezionare Commits (Commit). Scegliere l'ID del commit per cui si desidera visualizzare i commenti.

Viene visualizzata la pagina per il commit insieme con i commenti.

Aggiungi e rispondi ai commenti su un commit in un repository

È possibile utilizzare la CodeCommit console per aggiungere commenti al confronto tra un commit e un commit principale o al confronto tra due commit specificati. Puoi anche rispondere ai commenti con emoji, con i tuoi commenti o entrambi.

Aggiungi e rispondi ai commenti su un commit (console)

Puoi aggiungere e rispondere ai commenti a un commit con testo ed emoji. I tuoi commenti ed emoji sono contrassegnati come appartenenti all'utente o al ruolo IAM che hai usato per accedere alla console.

Per aggiungere e rispondere a commenti su un commit

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il repository in cui si desidera aggiungere un commento sui commit.
3. Nel riquadro di navigazione selezionare Commits (Commit). Scegliere l'ID del commit per cui si desidera aggiungere o rispondere ai commenti.

Viene visualizzata la pagina per il commit insieme con i commenti.

4. Per aggiungere un commento, procedere in uno dei seguenti modi:
 - Per aggiungere un commento generale, in Comments on changes (Commenti sulle modifiche), immettere il commento e quindi scegliere Save (Salva). È possibile utilizzare [Markdown](#) o immettere il commento in testo normale.

Comments on changes

New comment

☐ Preview markdown

[Learn more](#)

Did we also change the variable name in blf.py and concat.py?

Save

- Per aggiungere un commento a un file nel commit, individuare il nome del file. Scegliere Comment on file (Commenta file), immettere il commento e quindi selezionare Save (Salva).

ahs_count.py

Browse file contents

 Comment on file

New comment

☐ Preview markdown

[Learn more](#)

Save

Cancel

- Per aggiungere un commento a una riga modificata nel commit, andare alla riga in cui è presente la modifica. Scegliere l'icona del commento



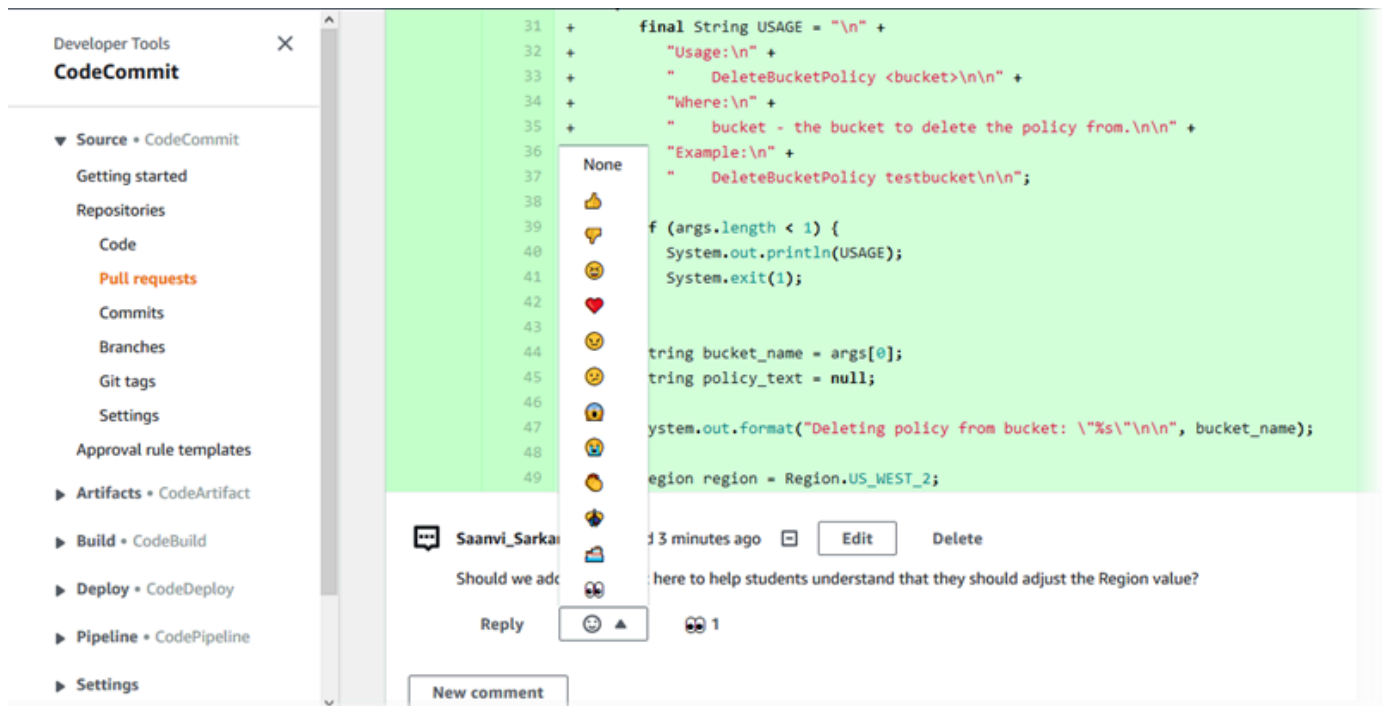
immettere il commento e quindi scegliere Save (Salva).



Note

Puoi modificare il commento dopo averlo salvato. Puoi anche eliminarne il contenuto. Il commento resterà con un messaggio che indica che il contenuto è stato eliminato. Provare a utilizzare la modalità Preview markdown (Markdown anteprima) per il commento prima di salvarlo.

5. Per rispondere a commenti su un commit, scegliere Reply (Rispondi). Per rispondere a un commento con un'emoji, scegli l'emoji che desideri dall'elenco. Puoi scegliere solo un'emoji per commento. Se desideri modificare la tua reazione emoji, scegline un'altra dall'elenco o scegli Nessuna per rimuovere la reazione.

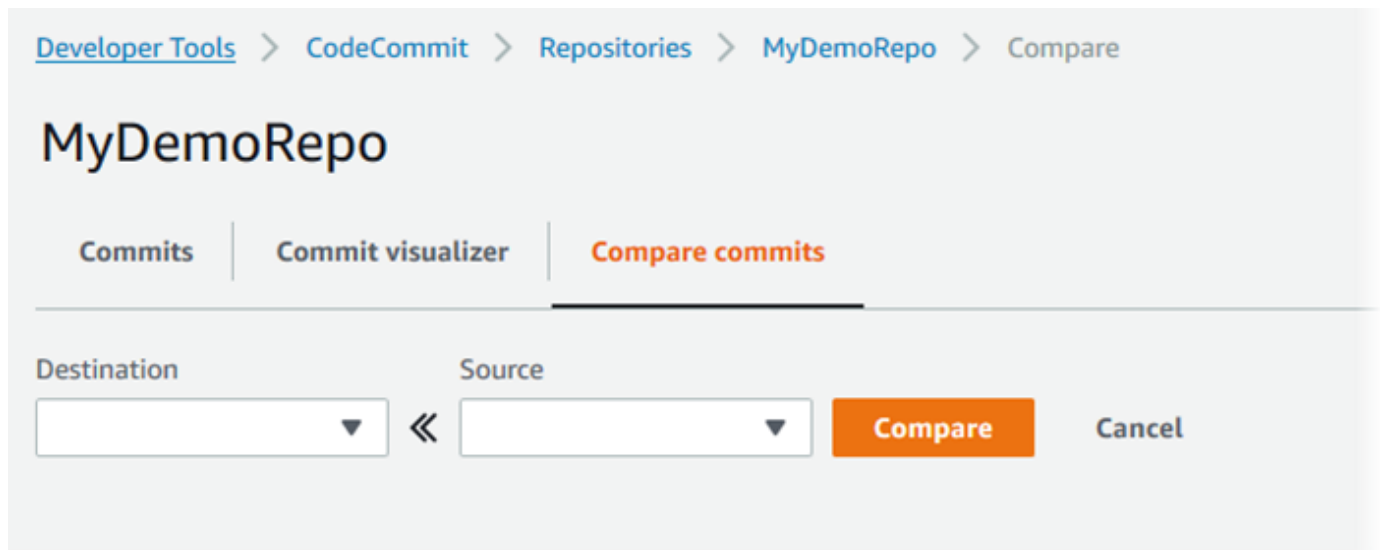


Aggiungi e rispondi ai commenti confrontando due specificatori di commit

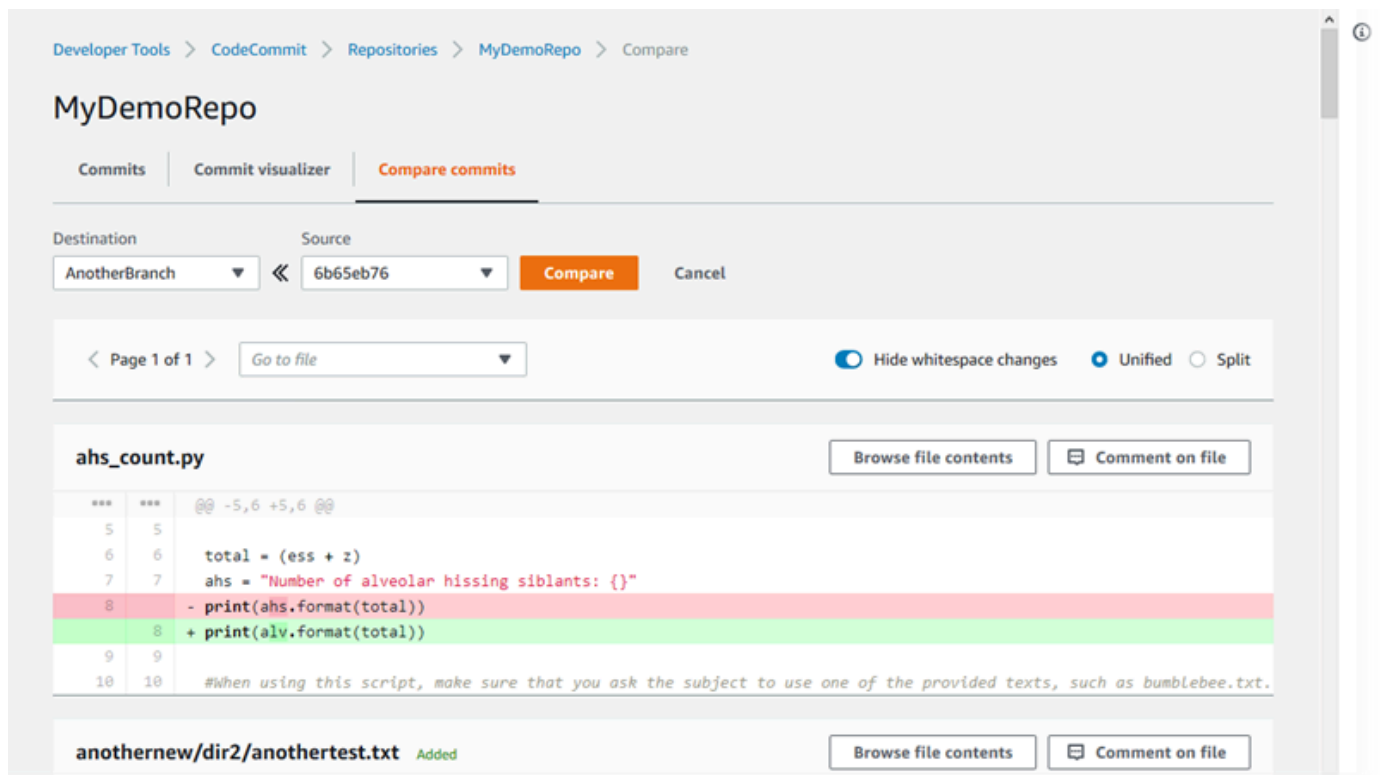
Puoi aggiungere commenti a un confronto tra rami, tag o commit.

Per aggiungere o rispondere a commenti per il confronto di specificatori di commit

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories (Repository), selezionare il repository in cui si desidera confrontare commit, rami o commit con tag.
3. Nel riquadro di navigazione scegliere Commits (Commit), quindi selezionare la scheda Compare commits (Confronta commit).



4. Usare i campi Destination (Destinazione) e Source (Origine) per confrontare due specificatori di commit. Usa gli elenchi a discesa o incolla il commit. IDs Scegli Confronta.



5. Effettuare una o più delle seguenti operazioni:
 - Per aggiungere commenti a file o righe, scegliere l'icona del commento



- Per aggiungere commenti generali sulle modifiche confrontate, andare a [Comments on changes](#) (Commenti sulle modifiche).

Visualizza, aggiungi, aggiorna e rispondi ai commenti (AWS CLI)

È possibile visualizzare, aggiungere, rispondere, aggiornare ed eliminare i contenuti di un commento eseguendo i seguenti comandi:

- Per visualizzare i commenti sul confronto tra due commit, eseguire [get-comments-for-compared-commit](#).
- Per visualizzare ulteriori informazioni su un commento, eseguire [get-comment](#).
- Per eliminare i contenuti di un commento creato, eseguire [delete-comment-content](#).
- Per creare un commento sul confronto tra due commit, eseguire [post-comment-for-compared-commit](#).
- Per aggiornare un commento, eseguire [update-comment](#).
- Per rispondere a un commento, [post-comment-reply](#) esegui.
- Per rispondere a un commento con un'emoji, corri [put-comment-reaction](#).
- Per visualizzare le reazioni emoji a un commento, esegui. [get-comment-reactions](#)

Come visualizzare i commenti su un commit

1. Eseguire il comando `get-comments-for-compared-commit` specificando:
 - Il nome del CodeCommit repository (con l'opzione `--repository-name`).
 - L'ID commit completo del commit after per stabilire la direzionalità del confronto (con l'opzione `--after-commit-id`).
 - L'ID commit completo del commit before per stabilire la direzionalità del confronto (con l'opzione `--before-commit-id`).
 - (Facoltativo) Un token di enumerazione per restituire il batch dei risultati successivo (con l'opzione `--next-token`).
 - (Facoltativo) Un numero intero non negativo per limitare il numero di risultati restituiti (con l'opzione `--max-results`).

Ad esempio, per visualizzare i commenti fatti sul confronto tra due commit in un repository chiamato *MyDemoRepo*:

```
aws codecommit get-comments-for-compared-commit --repository-name MyDemoRepo --  
before-commit-id 6e147360EXAMPLE --after-commit-id 317f8570EXAMPLE
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "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"
}

```

Per visualizzare i dettagli di un commento su un commit

1. Eseguire il comando `get-comment` specificando l'ID commento generato dal sistema. Esempio:

```
aws codecommit get-comment --comment-id ff30b348EXAMPLEb9aa670f
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```

{
  "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
      }
    }
  }
```

Per eliminare i contenuti di un commento su un commit

1. Eseguire il comando `delete-comment-content` specificando l'ID commento generato dal sistema.
Esempio:

```
aws codecommit delete-comment-content --comment-id ff30b348EXAMPLEb9aa670f
```

Note

Puoi eliminare il contenuto di un commento solo se hai applicato la `AWSCodeCommitFullAccess` politica o se hai l'`DeleteCommentContent` autorizzazione impostata su Consenti.

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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
      }
  }
}
```

Come creare un commento su un commit

1. Eseguire il comando `post-comment-for-compared-commit` specificando:

- Il nome del CodeCommit repository (con l'opzione `--repository-name`).
- L'ID commit completo del commit after per stabilire la direzionalità del confronto (con l'opzione `--after-commit-id`).
- L'ID commit completo del commit before per stabilire la direzionalità del confronto (con l'opzione `--before-commit-id`).
- Un token di idempotenza univoco generato dal client (con l'opzione `--client-request-token`).
- Il contenuto del commento (con l'opzione `--content`).
- Un elenco di informazioni sulla posizione in cui inserire il commento, tra cui:
 - Il nome del file che viene confrontato, inclusa l'estensione e la sottodirectory, se presente (con l'attributo `filePath`).
 - Il numero di riga della modifica in un file confrontato (con l'attributo `filePosition`).
 - Se il commento sulla modifica è before o after nel confronto tra i rami di origine e di destinazione (con l'attributo `relativeFileVersion`).

Ad esempio, per aggiungere il commento *"Can you add a test case for this?"* sulla modifica al `cl_sample.js` file nel confronto tra due commit in un repository denominato: *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. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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"
}
```

Come aggiornare un commento su un commit

1. Eseguire il comando `update-comment` specificando l'ID commento generato dal sistema e i contenuti con cui si desidera sostituire i contenuti esistenti.

Ad esempio, per aggiungere il contenuto *"Fixed as requested. I'll update the pull request."* a un commento con un ID di: *442b498bEXAMPLE5756813*

```
aws codecommit update-comment --comment-id 442b498bEXAMPLE5756813 --content "Fixed
as requested. I'll update the pull request."
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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
      }
  }
}
```

```
}
```

Per rispondere a un commento su un commit

1. Per pubblicare la risposta a un commento in una richiesta pull, eseguire il comando `post-comment-reply` specificando:
 - L'ID generato dal sistema del commento a cui si desidera rispondere (con l'opzione `--in-reply-to`).
 - Un token di idempotenza univoco generato dal client (con l'opzione `--client-request-token`).
 - Il contenuto della risposta (con l'opzione `--content`).

Ad esempio, per aggiungere la risposta *"Good catch. I'll remove them."* al commento con l'ID generato dal sistema di: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --  
content "Good catch. I'll remove them." --client-request-token 123Example
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "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": []  
  }  
}
```

Come rispondere a un commento su un commit con un'emoji

1. Per rispondere a un commento in una pull request con un'emoji o per modificare il valore della tua reazione emoji, esegui il `put-comment-reaction` comando, specificando:
 - L'ID generato dal sistema del commento a cui vuoi rispondere con un'emoji.
 - Il valore della reazione che desideri aggiungere o aggiornare. I valori accettabili includono emoji, codici brevi e valori Unicode supportati.

I seguenti valori sono supportati per gli emoji in: 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
	nessuno	U+0000

Ad esempio, per aggiungere l'emoji *:thumbsup:* al commento con l'ID generato dal sistema di: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit put-comment-reaction --comment-id abcd1234EXAMPLEb5678efgh --  
reaction-value :thumbsup:
```

2. Se il comando ha esito positivo, non produce alcun output.

Come visualizzare le reazioni emoji a un commento

1. Per visualizzare le reazioni emoji a un commento, inclusi gli utenti che hanno reagito con tali emoji, esegui il `get-comment-reactions` comando, specificando l'ID del commento generato dal sistema.

Ad esempio, per visualizzare le reazioni emoji al commento con l'ID generato dal sistema di: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit get-comment-reactions --comment-id abcd1234EXAMPLEb5678efgh
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "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/Li_Juan",  
        "arn:aws:iam::123456789012:user/Mary_Major",  
        "arn:aws:iam::123456789012:user/Jorge_Souza"  
      ]  
    }  
  ]  
}
```

```
        "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 un tag Git in AWS CodeCommit

È possibile utilizzare un tag Git per contrassegnare un commit con un'etichetta che aiuta gli altri utenti del repository a comprenderne l'importanza. Per creare un tag Git in un CodeCommit repository, puoi usare Git da un repository locale connesso al CodeCommit repository. Dopo aver creato un tag Git nel repository locale, puoi usarlo `git push --tags` per inviarlo al CodeCommit repository.

Per ulteriori informazioni, consulta [Visualizza i dettagli del tag](#).

Usa Git per creare un tag

Segui questi passaggi per utilizzare Git da un repository locale per creare un tag Git in un CodeCommit repository.

In questi passaggi, supponiamo che tu abbia già collegato il repository locale al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

1. Esegui il git tag ***new-tag-name commit-id*** comando, dove ***new-tag-name*** è il nome del nuovo tag Git e ***commit-id*** l'ID del commit da associare al tag Git.

Ad esempio, il comando seguente crea un tag Git denominato `beta` e lo associa all'ID commit `dc082f9a...af873b88`:

```
git tag beta dc082f9a...af873b88
```

2. Per inviare il nuovo tag Git dal repository locale al CodeCommit repository, esegui il `git push` **`remote-name new-tag-name`** comando, where **`remote-name`** è il nome del CodeCommit repository ed **`new-tag-name`** è il nome del nuovo tag Git.

Ad esempio, per inviare un nuovo tag Git denominato `beta` a un CodeCommit repository denominato `origin`:

```
git push origin beta
```

Note

Per inviare tutti i nuovi tag Git dal repository locale al CodeCommit repository, esegui. `git push --tags`

Per assicurarti che il tuo repository locale sia aggiornato con tutti i tag Git nel CodeCommit repository, esegui `git fetch` seguito da. `git fetch --tags`

Per altre opzioni, consulta la documentazione di Git.

Visualizza i dettagli del tag Git in AWS CodeCommit

In Git, un tag è un'etichetta che è possibile applicare a un riferimento come un commit per contrassegnarlo con le informazioni che potrebbero essere importanti per altri utenti del repository. Ad esempio, è possibile assegnare un tag al commit che era il punto di rilascio beta di un progetto con il tag **`beta`**. Per ulteriori informazioni, consulta [Usa Git per creare un tag](#). I tag Git sono diversi dai tag del repository. Per ulteriori informazioni su come usare i tag del repository, consulta [Aggiunta di un tag a un repository](#).

Puoi usare la AWS CodeCommit console per visualizzare le informazioni sui tag Git nel tuo repository, inclusi la data e il messaggio di commit del commit a cui fa riferimento ogni tag Git. Utilizzando la console, è possibile confrontare il commit contrassegnato dal tag con il commit

principale del ramo predefinito nel proprio repository. Come per qualsiasi altro commit, è anche possibile visualizzare il codice nel punto di tale tag Git.

Puoi anche usare Git dal tuo terminale o dalla riga di comando per visualizzare i dettagli sui tag Git in un repository locale.

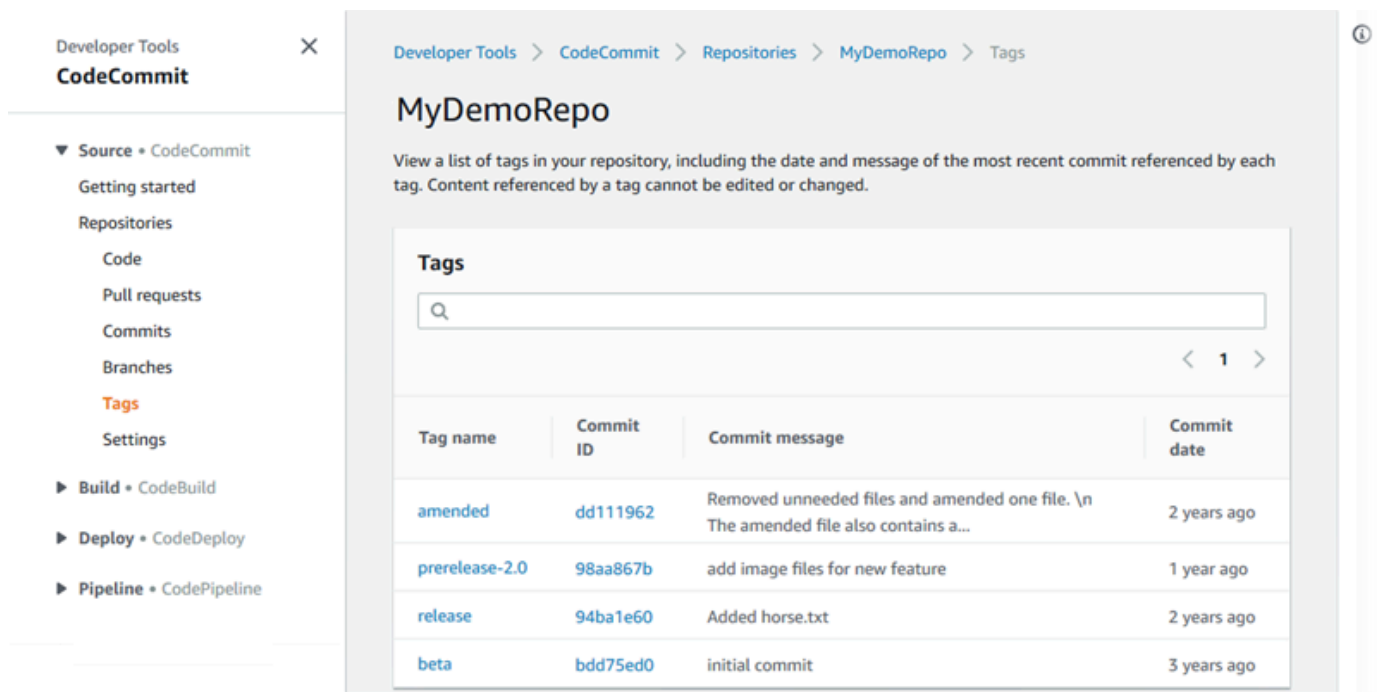
Argomenti

- [Visualizza i dettagli dei tag \(console\)](#)
- [Visualizza i dettagli del tag Git \(Git\)](#)

Visualizza i dettagli dei tag (console)

Usa la AWS CodeCommit console per visualizzare rapidamente un elenco di tag Git per il tuo repository e i dettagli sui commit a cui fanno riferimento i tag Git.

1. [Apri la CodeCommit console su codecommit/home](https://console.aws.amazon.com/codesuite/). <https://console.aws.amazon.com/codesuite/>
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera visualizzare i tag.
3. Nel riquadro di navigazione, scegliere Git tags (Tag 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 'Source', 'Build', 'Deploy', and 'Pipeline'. The 'Source' section is expanded, showing 'Getting started', 'Repositories', 'Code', 'Pull requests', 'Commits', 'Branches', 'Tags' (highlighted), and 'Settings'. The main content area shows the 'Tags' page for a repository named 'MyDemoRepo'. It includes a search bar and a table of tags.

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. Esegui una delle seguenti operazioni:
 - Per visualizzare il codice così com'era al commit, selezionare il nome del tag Git.

- Per visualizzare i dettagli del commit, tra cui il messaggio di commit completo, il committer e l'autore, scegliere l'ID di commit abbreviato.

Visualizza i dettagli del tag Git (Git)

Per usare Git per visualizzare i dettagli sui tag Git in un repository locale, esegui uno dei seguenti comandi:

- [git tag](#) per visualizzare una lista dei nomi dei tag Git.
- [git show](#) per visualizzare informazioni su un tag Git specifico.
- [git ls-remote](#) per visualizzare informazioni sui tag Git in un CodeCommit repository.

Note

Per assicurarti che il tuo repository locale sia aggiornato con tutti i tag Git presenti nel CodeCommit repository, esegui `git fetch` seguito da `git fetch --tags`

Nei passaggi seguenti, supponiamo che tu abbia già collegato il repository locale a un repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

Per visualizzare un elenco di tag Git in un repository locale

1. Eseguire il comando `git tag`:

```
git tag
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
beta  
release
```

Note

Se non è stato definito alcun tag, `git tag` non restituisce alcun valore.

Per altre opzioni, consulta la documentazione di Git.

Per visualizzare informazioni su un tag Git in un repository locale

1. Esegui il comando `git show tag-name`. Ad esempio, per visualizzare informazioni su un tag Git denominato `beta`, eseguire:

```
git show beta
```

2. Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
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

Per uscire dell'output delle informazioni del tag Git, digitare `:q`.

Per altre opzioni, consulta la documentazione di Git.

Per visualizzare informazioni sui tag Git in un CodeCommit repository

1. Esegui il comando `git ls-remote --tags`.

```
git ls-remote --tags
```

2. In caso di successo, questo comando produce come output un elenco dei tag Git nel CodeCommit repository:

```
129ce87a...70fbffba    refs/tags/beta
785de9bd...59b402d8    refs/tags/release
```

Se non sono stati definiti tag Git, `git ls-remote --tags` restituisce una riga vuota.

Per altre opzioni, consulta la documentazione di Git.

Eliminare un tag Git in AWS CodeCommit

Per eliminare un tag Git in un CodeCommit repository, usa Git da un repository locale connesso al CodeCommit repository.

Usa Git per eliminare un tag Git

Segui questi passaggi per utilizzare Git da un repository locale per eliminare un tag Git in un CodeCommit repository.

Questi passaggi sono scritti partendo dal presupposto che tu abbia già collegato il repository locale al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

1. Per eliminare il tag Git dal repository locale, esegui il `git tag -d tag-name` comando where **tag-name** è il nome del tag Git che desideri eliminare.

Tip

Per ottenere un elenco di nomi di tag Git, eseguire `git tag`.

Ad esempio, per eliminare un tag Git nel repository locale denominato `beta`:

```
git tag -d beta
```

2. Per eliminare il tag Git dal CodeCommit repository, esegui il `git push remote-name --delete tag-name` comando dove **remote-name** è il soprannome che il repository locale usa per il CodeCommit repository ed **tag-name** è il nome del tag Git che desideri eliminare dal repository. CodeCommit

 Tip

Per ottenere un elenco dei nomi dei CodeCommit repository e relativi URLs, esegui il comando. `git remote -v`

Ad esempio, per eliminare un tag Git denominato `beta` nel CodeCommit repository denominato `origin`:

```
git push origin --delete beta
```

Lavorare con le filiali nei repository AWS CodeCommit

Cos'è un ramo? In Git, i rami sono puntatori o riferimenti a un commit. Nello sviluppo, sono un modo utile di organizzare il lavoro. Puoi usare i rami per separare il lavoro su una versione nuova o diversa dei file senza influire sul lavoro in altri rami. Puoi utilizzare i rami per sviluppare nuove caratteristiche, archiviare una versione specifica del progetto a partire da un particolare commit e molto altro. Quando crei il tuo primo commit, viene creato automaticamente un ramo predefinito. Questo ramo predefinito è quello utilizzato come ramo di base o predefinito nei repository locali (repository) quando gli utenti clonano il repository. Il nome di quel ramo predefinito varia a seconda di come si crea il primo commit. Se aggiungi il primo file al tuo repository utilizzando la CodeCommit console, la AWS CLI o una delle due SDKs, il nome di quel ramo predefinito è `main`. Questo è il nome di ramo predefinito utilizzato negli esempi di questa guida. Se esegui il push del primo commit utilizzando un client Git, il nome del ramo predefinito è quello specificato dal client Git come predefinito. Prendi in considerazione la possibilità di configurare il tuo client Git per utilizzare `main` come nome per il ramo iniziale.

In CodeCommit, puoi cambiare il ramo predefinito per il tuo repository. Puoi inoltre creare ed eliminare rami e visualizzare i dettagli su un ramo. Puoi confrontare rapidamente le differenze tra un ramo e il ramo predefinito (o tra due rami qualsiasi). Per visualizzare la cronologia dei rami e delle unioni nel repository, è possibile utilizzare il [visualizzatore Commit](#), illustrato nell'immagine seguente.

The screenshot shows the AWS CodeCommit interface for a repository named 'MyDemoRepo'. The left sidebar contains a 'Developer Tools' section with a search icon and a list of options: Source (selected), CodeCommit, Getting started, Repositories, Code, Pull requests, Commits, Branches, Tags, Settings, Build (CodeBuild), Deploy (CodeDeploy), and Pipeline (CodePipeline). The main content area has a breadcrumb trail: Developer Tools > CodeCommit > Repositories > MyDemoRepo > Commits. Below this is a tabbed interface with 'Commits', 'Commit visualizer' (active), and 'Compare commits'. The 'Commit visualizer' tab shows a commit history with a visual timeline on the left and a list of commits on the right. The commits are listed in descending order of time, with the most recent at the top.

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
edacdfce	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

Per informazioni sull'utilizzo di altri aspetti del repository in CodeCommit, consulta [Lavorare con i repository](#),, e. [Lavorare con i file](#) [Utilizzo delle richieste pull](#) [Lavorare con i commit](#) [Utilizzo delle preferenze dell'utente](#)

Argomenti

- [Crea un ramo in AWS CodeCommit](#)
- [Limita i push e le unioni ai rami in AWS CodeCommit](#)
- [Visualizza i dettagli della filiale in AWS CodeCommit](#)
- [Confronta e unisci i rami in AWS CodeCommit](#)
- [Modifica le impostazioni del ramo in AWS CodeCommit](#)
- [Elimina un ramo in AWS CodeCommit](#)

Crea un ramo in AWS CodeCommit

È possibile utilizzare la CodeCommit console o creare rami per il repository. AWS CLI Questo è un modo rapido per separare il lavoro su una versione dei file nuova o diversa senza influire sul lavoro nel ramo predefinito. Dopo aver creato un ramo nella CodeCommit console, devi inserire la modifica nel repository locale. In alternativa, puoi creare un ramo localmente e quindi utilizzare Git da un repository locale connesso al CodeCommit repository per inviare tale modifica.

Argomenti

- [Crea un ramo \(console\)](#)
- [Crea un ramo \(Git\)](#)
- [Crea un ramo \(AWS CLI\)](#)

Crea un ramo (console)

È possibile utilizzare la CodeCommit console per creare un ramo in un CodeCommit repository. Gli utenti vedranno il nuovo ramo la prossima volta che eseguono il pull delle modifiche dal repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera creare un ramo.
3. Nel riquadro di navigazione selezionare Branches (Rami).
4. Selezionare Crea branch (Crea ramo).

Create branch [X]

Branch name
feature_advanced-class

Branch from
[Dropdown menu]

Search: [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

Immettere un nome per il ramo in Branch name (Nome ramo). In Branch from (Ramo da), scegliere un ramo o un tag dall'elenco o incollare un ID commit. Selezionare Crea branch (Crea ramo).

Crea un ramo (Git)

Segui questi passaggi per utilizzare Git da un repository locale per creare un ramo in un repository locale e quindi inviare quel ramo al CodeCommit repository.

Questi passaggi sono scritti partendo dal presupposto che tu abbia già collegato il repository locale al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

1. Crea un ramo nel tuo repository locale eseguendo il git checkout -b **new-branch-name** comando, **new-branch-name** dov'è il nome del nuovo ramo.

Ad esempio, il comando seguente crea un ramo denominato MyNewBranch nel repository locale:

```
git checkout -b MyNewBranch
```

2. Per inviare il nuovo ramo dal repository locale al CodeCommit repository, esegui il git push comando, specificando sia il che. **remote-name new-branch-name**

Ad esempio, per inserire un nuovo ramo nel repository locale denominato nel CodeCommit repository con MyNewBranch il soprannome: origin

```
git push origin MyNewBranch
```

Note

Se aggiungi l'-uopzione a git push (ad esempio,git push -u origin main), in futuro potrai farne a git push meno **remote-namebranch-name**. Vengono impostate le informazioni di registrazione upstream. Per ottenere le informazioni di registrazione upstream, eseguire git remote show **remote-name** (ad esempio, git remote show origin).

Per vedere l'elenco di tutti i rami di registrazione remoti e locali, eseguire git branch --all.

Per configurare un ramo nel repository locale collegato a un ramo del CodeCommit repository, esegui. git checkout **remote-branch-name**

Per altre opzioni, consulta la documentazione di Git.

Crea un ramo (AWS CLI)

Per utilizzare AWS CLI i comandi con CodeCommit, installa AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Segui questi passaggi per AWS CLI creare un ramo in un CodeCommit repository e poi invia quel ramo al CodeCommit repository. Per i passaggi per creare un commit iniziale e specificare il nome del ramo predefinito per un repository vuoto, consulta [Creare il primo commit per un repository](#) utilizzando il. AWS CLI

1. Eseguire il comando create-branch specificando:

- Il nome del CodeCommit repository in cui viene creato il ramo (con l'--repository-nameopzione).

 Note

Per ottenere il nome del CodeCommit repository, esegui il comando [list-repositories](#).

- Il nome del nuovo ramo (con l'opzione --branch-name).
- L'ID del commit a cui punta il nuovo ramo (con l'opzione --commit-id).

Ad esempio, per creare un ramo denominato MyNewBranch che punti a commit ID 317f8570EXAMPLE in un repository denominato: CodeCommit MyDemoRepo

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch  
--commit-id 317f8570EXAMPLE
```

Questo comando restituisce un output solo se ci sono errori.

2. Per aggiornare l'elenco dei rami del CodeCommit repository disponibili nel repository locale con il nuovo nome del ramo remoto, esegui. git remote update **remote-name**

Ad esempio, per aggiornare l'elenco dei rami disponibili per il CodeCommit repository con il soprannome: origin

```
git remote update origin
```

 Note

In alternativa, è possibile eseguire il comando git fetch. Puoi anche visualizzare tutti i rami remoti eseguendogit branch --all, ma finché non aggiorni l'elenco del repository locale, il ramo remoto che hai creato non appare nell'elenco.

Per altre opzioni, consulta la documentazione di Git.

3. Per configurare un ramo nel repository locale connesso al nuovo ramo nel CodeCommit repository, esegui. `git checkout remote-branch-name`

Note

Per ottenere un elenco dei nomi dei CodeCommit repository e relativi URLs, esegui il comando. `git remote -v`

Limita i push e le unioni ai rami in AWS CodeCommit

Per impostazione predefinita, qualsiasi utente del CodeCommit repository che dispone di autorizzazioni sufficienti per inviare codice al repository può contribuire a qualsiasi ramo del repository. Questo è vero indipendentemente da come aggiungi un ramo al repository: utilizzando la console, la riga di comando o Git. Tuttavia, è possibile configurare un ramo in modo tale che solo alcuni utenti del repository possano inserire o eseguire il push del codice in quel ramo. Ad esempio, potresti configurare un ramo utilizzato per il codice di produzione in modo che solo un sottoinsieme di sviluppatori senior possa eseguire il push o unire le modifiche a tale ramo. Gli altri sviluppatori possono ancora estrarre dal ramo, creare propri rami e creare richieste pull, ma non possono eseguire push o unioni delle modifiche a quel ramo. Puoi configurare questo accesso creando una policy condizionale che utilizza una chiave di contesto per una o più filiali in IAM.

Note

Per completare alcune delle procedure descritte in questo argomento, devi accedere con un utente amministrativo che disponga di autorizzazioni sufficienti per configurare e applicare le politiche IAM. Per ulteriori informazioni, consulta [Creazione di un utente e un gruppo di amministratori IAM](#).

Argomenti

- [Configura una policy IAM per limitare i push e le unioni a una filiale](#)
- [Applica la policy IAM a un gruppo o ruolo IAM](#)
- [Prova la policy](#)

Configura una policy IAM per limitare i push e le unioni a una filiale

Puoi creare una policy in IAM che impedisca agli utenti di aggiornare una filiale, incluso l'invio di commit a una filiale e l'unione delle richieste pull a una filiale. A tale scopo, la policy utilizza un'istruzione condizionale, in modo che l'effetto dell'istruzione Deny si applichi solo se la condizione è soddisfatta. Le azioni APIs che includi nella Deny dichiarazione determinano quali azioni non sono consentite. Puoi configurare questa politica in modo che si applichi a un solo ramo in un repository, a più rami in un repository o a tutti i rami che soddisfano i criteri in tutti i repository di un account Amazon Web Services.

Per creare una policy condizionale per i rami

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel riquadro di navigazione, scegli Policy.
3. Scegli Create Policy (Crea policy).
4. Scegliere JSON e incollare la policy di esempio seguente. Sostituire il valore di Resource con l'ARN del repository che contiene il ramo per cui si desidera limitare l'accesso. Sostituire il valore di `codecommit:References` con un riferimento al ramo o ai rami per cui si desidera limitare l'accesso. Ad esempio, questa politica vieta di inviare commit, unire rami, eliminare rami, eliminare file, unire richieste pull e aggiungere file a un ramo denominato `main` e a un ramo denominato in un repository denominato `main: prod 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"
        }
    }
}
]
}

```

I rami in Git sono semplicemente puntatori (riferimenti) al valore SHA-1 del commit di testa ed è per questo che la condizione usa References. L'istruzione Null è obbligatoria in ogni policy il cui effetto è Deny e dove GitPush è una delle operazioni. Questo è necessario a causa del modo in cui Git e git-receive-pack funzionano quando si effettua il push delle modifiche da un repository locale a CodeCommit.

Tip

Per creare una politica che si applichi a tutte le filiali denominate main in tutti i repository di un account Amazon Web Services, modifica il valore di Resource da un repository ARN a un asterisco (*).

5. Scegliere Esamina policy. Correggere gli eventuali errori nell'istruzione di policy e quindi continuare con Create policy (Crea policy).
6. Quando il JSON è convalidato, viene visualizzata la pagina Create policy (Crea policy). Viene visualizzato un avviso nella sezione Summary (Riepilogo) indicante che questa policy non concede autorizzazioni. Si tratta di un comportamento normale.
 - In Name (Nome), immettere un nome per la policy, ad esempio **DenyChangesToMain**.

- In Description (Descrizione), immettere una descrizione dello scopo della policy. Questo passaggio è facoltativo, ma è consigliato.
- Scegli Crea policy.

Applica la policy IAM a un gruppo o ruolo IAM

Hai creato una policy che limita i push e le fusioni a una filiale, ma la policy non ha effetto finché non la applichi a un utente, gruppo o ruolo IAM. Come best practice, valuta la possibilità di applicare la policy a un gruppo o a un ruolo IAM. L'applicazione delle policy ai singoli utenti IAM non offre una scalabilità adeguata.

Per applicare la policy condizionale a un gruppo o un ruolo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, se desideri applicare la policy a un gruppo IAM, scegli Gruppi. Se desideri applicare la policy a un ruolo assunto dagli utenti, scegli Ruolo. Scegliere il nome del gruppo o del ruolo.
3. Nella scheda Permissions (Autorizzazioni), scegli Attach Policy (Associa policy).
4. Selezionare la policy condizionale creata dall'elenco delle policy e quindi scegliere Attach policy (Collega policy).

Per ulteriori informazioni, consulta [Allegare e scollegare le politiche IAM](#).

Prova la policy

È necessario testare gli effetti della policy che hai applicato al gruppo o al ruolo per assicurarti che funzioni come previsto. Esistono molti modi per eseguire questa operazione. Ad esempio, per testare una policy simile a quella mostrata sopra, è possibile:

- Accedi alla CodeCommit console con un utente IAM che è membro di un gruppo IAM a cui è stata applicata la policy o che assume un ruolo a cui è applicata la policy. Nella console aggiungi un file nel ramo in cui si applicano le restrizioni. Viene visualizzato un messaggio di errore quando tenti di salvare o caricare un file nel ramo. Aggiungi un file a un ramo diverso. L'operazione è completata.
- Accedi alla CodeCommit console con un utente IAM che è membro di un gruppo IAM a cui è stata applicata la policy o che assume un ruolo a cui è applicata la policy. Creare una richiesta di pull che

viene unita al ramo in cui si applicano le restrizioni. Potrai creare la richiesta di pull, ma riceverai un messaggio di errore se la tenti di unire.

- Dal terminale o dalla riga di comando, crea un commit sul ramo in cui si applicano le restrizioni, quindi trasferiscilo nel CodeCommit repository. Viene visualizzato un messaggio di errore. I commit e i push effettuati da altri rami funzionano normalmente.

Visualizza i dettagli della filiale in AWS CodeCommit

È possibile utilizzare la CodeCommit console per visualizzare i dettagli sulle filiali in un CodeCommit repository. Puoi visualizzare la data dell'ultimo commit a un ramo, il messaggio di commit e molto altro. Puoi anche usare Git AWS CLI o Git da un repository locale connesso al CodeCommit repository.

Argomenti

- [Visualizza i dettagli della filiale \(console\)](#)
- [Visualizza i dettagli della filiale \(Git\)](#)
- [Visualizza i dettagli della filiale \(AWS CLI\)](#)

Visualizza i dettagli della filiale (console)

Usa la CodeCommit console per visualizzare rapidamente un elenco di filiali per il tuo repository e i dettagli sulle filiali.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository di cui si desidera visualizzare i dettagli dei rami.
3. Nel riquadro di navigazione selezionare Branches (Rami).

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Branches

MyDemoRepo

Branches Info Delete branch View branch View last commit Create pull request Create branch

Q

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

- Il nome del ramo usato come predefinito per il repository viene visualizzato accanto a Ramo predefinito. Per visualizzare i dettagli relativi al più recente commit a un ramo, scegliere il ramo, quindi scegliere View last commit (Visualizza ultimo commit). Per visualizzare i file e il codice in un ramo, scegli il nome del ramo.

Visualizza i dettagli della filiale (Git)

Per utilizzare Git da un repository locale per visualizzare i dettagli sui rami di tracciamento locali e remoti di un CodeCommit repository, esegui il git branch comando.

I passaggi seguenti sono scritti partendo dal presupposto che il repository locale sia già stato collegato al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

- Esegui il comando git branch e specifica l'opzione --all.

```
git branch --all
```

- Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
MyNewBranch
* main
remotes/origin/MyNewBranch
```



```
remotes/origin/main
```

Accanto al ramo al momento aperto viene visualizzato un asterisco (*). Le voci che seguono sono riferimenti di registrazione remota.

 Tip

git branch mostra i rami locali.

git branch -r mostra i rami remoti.

git checkout **existing-branch-name** passa al nome ramo specificato e, se git branch viene eseguito immediatamente dopo, lo visualizza con un asterisco (*).

git remote update **remote-name** aggiorna il repository locale con l'elenco dei rami del repository disponibili. CodeCommit (Per ottenere un elenco dei nomi dei CodeCommit repository e relativi URLs, esegui il git remote -v comando.)

Per altre opzioni, consulta la documentazione di Git.

Visualizza i dettagli della filiale (AWS CLI)

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per utilizzare il AWS CLI per visualizzare i dettagli sui rami in un CodeCommit repository, esegui uno o più dei seguenti comandi:

- Per visualizzare un elenco di nomi di rami, esegui [list-branches](#).
- [Per visualizzare informazioni su un ramo specifico, esegui get-branch.](#)

Come visualizzare un elenco di nomi delle ramificazioni

1. Esegui il list-branches comando, specificando il nome del CodeCommit repository (con l'opzione). --repository-name

 Tip

[Per ottenere il nome del repository, esegui il comando CodeCommit list-repositories.](#)

Ad esempio, per visualizzare i dettagli sui rami in un repository denominato: CodeCommit MyDemoRepo

```
aws codecommit list-branches --repository-name MyDemoRepo
```

2. In caso di successo, l'output del comando è un oggetto `branchNameList`, con una voce per ogni ramo.

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "branches": [
    "MyNewBranch",
    "main"
  ]
}
```

Per visualizzare informazioni su un ramo

1. Eseguire il comando `get-branch` specificando:
 - Il nome del repository (con l'opzione `--repository-name`).
 - Il nome del ramo (con l'opzione `--branch-name`).

Ad esempio, per visualizzare informazioni su un ramo denominato `MyNewBranch` in un CodeCommit repository denominato: `MyDemoRepo`

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

2. In caso di successo, l'output di questo comando è il nome del ramo e l'ID dell'ultimo commit effettuato per il ramo.

Questo è un esempio di output in base al comando di esempio precedente:

```
{
  "branch": {
    "branchName": "MyNewBranch",
    "commitID": "317f8570EXAMPLE"
  }
}
```

```
}  
}
```

Confronta e unisci i rami in AWS CodeCommit

È possibile utilizzare la CodeCommit console per confrontare le filiali in un CodeCommit repository. Il confronto dei rami consente di visualizzare rapidamente le differenze tra un ramo e il ramo predefinito o visualizzare le differenze tra due rami qualsiasi.

Argomenti

- [Confronta un ramo con il ramo predefinito](#)
- [Confronta due rami specifici](#)
- [Unisci due rami \(\)AWS CLI](#)

Confronta un ramo con il ramo predefinito

Usa la CodeCommit console per visualizzare rapidamente le differenze tra un ramo e il ramo predefinito del tuo repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera confrontare i rami.
3. Nel riquadro di navigazione scegliere Commits (Commit), quindi selezionare la scheda Compare commits (Confronta commit).
4. In Destination (Destinazione), scegliere il nome del ramo predefinito. In Source (Origine), scegliere il ramo da confrontare con il ramo predefinito. Scegli Confronta.

Confronta due rami specifici

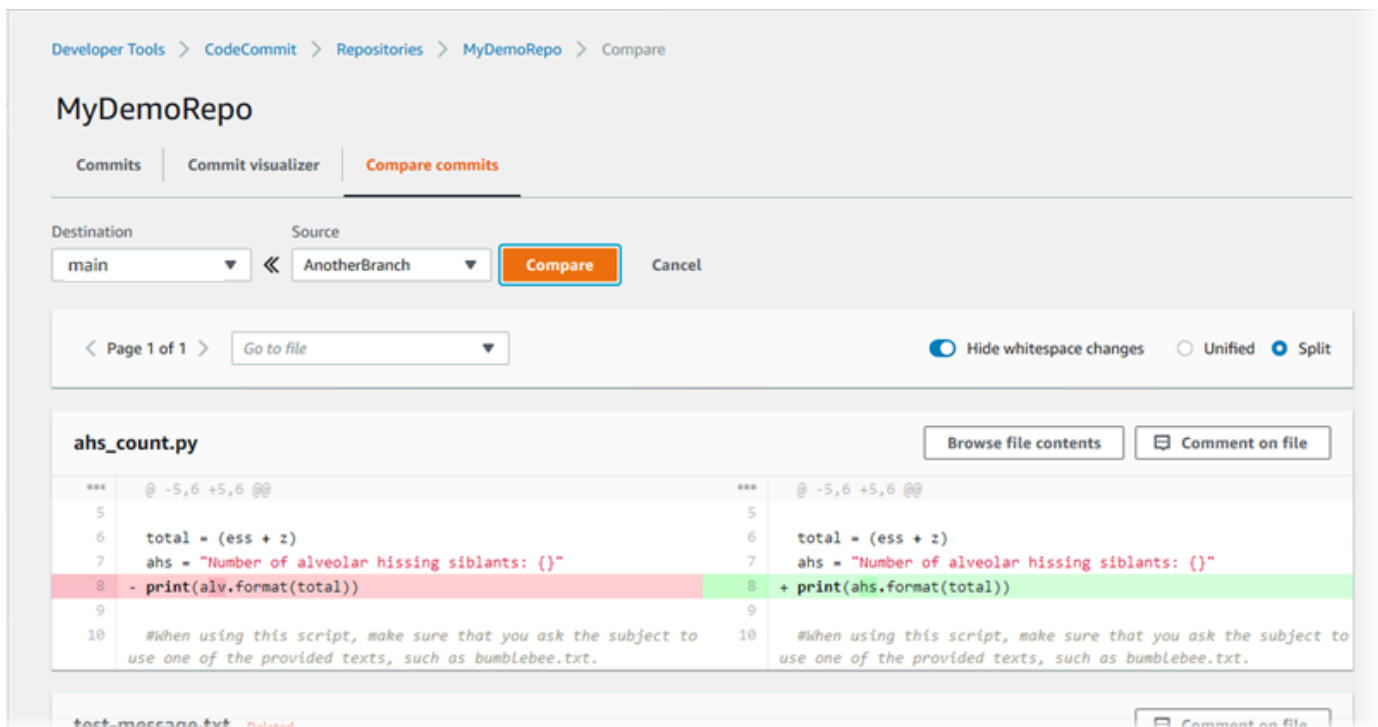
Usa la CodeCommit console per visualizzare le differenze tra due rami che desideri confrontare.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository in cui si desidera confrontare i rami.

3. Nel riquadro di navigazione scegliere Commits (Commit), quindi selezionare la scheda Compare commits (Confronta commit).
4. In Destination (Destinazione) e Source (Origine), scegliere i due rami da confrontare e quindi scegliere Compare (Confronta). Per visualizzare l'elenco dei file modificati, espandere l'elenco dei file modificati. È possibile visualizzare le modifiche nei file affiancati (visualizzazione doppia) o inline (visualizzazione unificata).

Note

Se hai effettuato l'accesso come utente IAM, puoi configurare e salvare le tue preferenze per la visualizzazione del codice e altre impostazioni della console. Per ulteriori informazioni, consulta [Utilizzo delle preferenze dell'utente](#).



Unisci due rami ()AWS CLI

È possibile unire due rami in un CodeCommit repository utilizzando una delle strategie di unione disponibili eseguendo uno dei seguenti comandi: AWS CLI

- Per unire due rami utilizzando la strategia di unione avanzamento rapido, esegui il comando [merge-branches-by-fast-forward](#).
- Per unire due rami utilizzando la strategia di unione squash, esegui il comando [merge-branches-by-squash](#).
- Per unire due rami utilizzando la strategia di unione a tre vie, esegui il comando [merge-branches-by-three-way](#).

Puoi anche testare le unioni eseguendo il comando `create-unreferenced-merge-commit`. Per ulteriori informazioni, consulta [Risolvere i conflitti in una richiesta di pull](#).

Note

Per utilizzare AWS CLI i comandi con CodeCommit, installa il. AWS CLI Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Da utilizzare AWS CLI per unire due rami in un repository CodeCommit

1. Per unire due rami utilizzando la strategia di unione avanzamento rapido, esegui il comando `merge-branches-by-fast-forward` specificando:

- Il nome del ramo di origine che contiene le modifiche che desideri unire (con l'opzione `--source-commit-specifier`).
- Il nome del ramo di destinazione in cui desideri unire le modifiche (con l'opzione `--destination-commit-specifier`).
- Il nome del repository (con l'opzione `--repository-name`).

Ad esempio, per unire un ramo di origine denominato *bugfix-1234* in un ramo di destinazione denominato in un *preprod* repository denominato: *MyDemoRepo*

```
aws codecommit merge-branches-by-fast-forward --source-commit-specifier bugfix-  
bug1234 --destination-commit-specifier preprod --repository-name MyDemoRepo
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
```

```
"commitId": "4f178133EXAMPLE",  
"treeId": "389765daEXAMPLE"  
}
```

2.

Per unire due rami utilizzando la strategia di unione squash, esegui il comando `merge-branches-by-squash` specificando:

- Il nome del ramo di origine che contiene le modifiche che desideri unire (con l'opzione `--source-commit-specifier`).
- Il nome del ramo di destinazione in cui desideri unire le modifiche (con l'opzione `--destination-commit-specifier`).
- Il nome del repository (con l'opzione `--repository-name`).
- Il messaggio di commit da includere (con l'opzione `--commit-message`).
- Il nome da utilizzare per il commit (con l'opzione `--name`).
- L'indirizzo e-mail da utilizzare per il commit (con l'opzione `--email`).

Ad esempio, per unire un ramo di origine denominato *bugfix-bug1234* con un ramo di destinazione denominato *bugfix-quarterly* in un repository denominato: *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
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

3.

Per unire due rami utilizzando la strategia di unione a tre vie, esegui il comando `merge-branches-by-three-way` specificando:

- Il nome del ramo di origine che contiene le modifiche che desideri unire (con l'opzione `--source-commit-specifier`).

- Il nome del ramo di destinazione in cui desideri unire le modifiche (con l'opzione `--destination-commit-specifier`).
- Il nome del repository (con l'opzione `--repository-name`).
- Il messaggio di commit da includere (con l'opzione `--commit-message`).
- Il nome da utilizzare per il commit (con l'opzione `--name`).
- L'indirizzo e-mail da utilizzare per il commit (con l'opzione `--email`).

Ad esempio, per unire un ramo di origine denominato *main* con un ramo di destinazione denominato *bugfix-1234* in un repository denominato: *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
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Modifica le impostazioni del ramo in AWS CodeCommit

È possibile modificare il ramo da utilizzare come ramo predefinito nella AWS CodeCommit console o con AWS CLI. Ad esempio, se hai creato il tuo primo commit utilizzando un client Git che imposta il ramo predefinito su master, puoi creare un ramo denominato *main* e quindi modificare le impostazioni del ramo in modo che il nuovo ramo sia impostato come ramo predefinito per il repository. Per modificare altre impostazioni del ramo, puoi usare Git da un repository locale collegato al CodeCommit repository.

Argomenti

- [Cambia il ramo predefinito \(console\)](#)
- [Cambia il ramo predefinito \(AWS CLI\)](#)

Cambia il ramo predefinito (console)

È possibile specificare quale ramo è il ramo predefinito in un CodeCommit repository nella AWS CodeCommit console.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository), selezionare il nome del repository per cui si desidera modificare le impostazioni.
3. Nel pannello di navigazione scegli Impostazioni.
4. In Default branch (Ramo predefinito), scegliere l'elenco a discesa dei rami e quindi un altro ramo. Scegli Save (Salva).

Tip

- Se non vedi un altro ramo nell'elenco a discesa, significa che non hai creato nessun ramo aggiuntivo. Non è possibile modificare il ramo predefinito di un repository se il repository ha un solo ramo. Per ulteriori informazioni, consulta [Crea un ramo in AWS CodeCommit](#).
- Se non vedi la sezione Ramo predefinito ma vedi invece gli elementi per le regole e le connessioni di notifica, ti trovi nel menu delle impostazioni generali della console. Il menu delle impostazioni per i repository è elencato in Repository allo stesso livello delle richieste Code e Pull.

Cambia il ramo predefinito ()AWS CLI

Per utilizzare AWS CLI i comandi con CodeCommit, installa il AWS CLI. Per ulteriori informazioni, consulta [Guida di riferimento alla riga di comando](#).

Per utilizzare il comando AWS CLI per modificare le impostazioni del ramo di un repository in un CodeCommit repository, esegui il comando seguente:

- [update-default-branch](#) per modificare il ramo predefinito.

Per modificare il ramo predefinito

1. Eseguire il comando update-default-branch specificando:

- Il nome del CodeCommit repository in cui viene aggiornato il ramo predefinito (con l'--repository-nameopzione).

 Tip

Per ottenere il nome del CodeCommit repository, esegui il comando [list-repositories](#).

- Il nome del nuovo ramo predefinito (con l'opzione --default-branch-name).

 Tip

[Per ottenere il nome del ramo, esegui il comando list-branches.](#)

2. Ad esempio, per modificare il ramo predefinito MyNewBranch in un CodeCommit repository denominato: MyDemoRepo

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Questo comando restituisce un output solo se ci sono errori.

Per altre opzioni, consulta la documentazione di Git.

Elimina un ramo in AWS CodeCommit

È possibile utilizzare la CodeCommit console per eliminare un ramo in un repository. L'eliminazione di un branch in CodeCommit non elimina quel ramo in un repository locale, quindi gli utenti potrebbero continuare ad avere copie di quel branch fino alla prossima modifica. Per eliminare un ramo localmente e inviare la modifica al CodeCommit repository, usa Git da un repository locale connesso al CodeCommit repository.

L'eliminazione di un ramo non elimina i commit. Vengono però eliminati tutti i riferimenti ai commit nel ramo in questione. Se si elimina un ramo che contiene commit che non sono stati uniti in un altro ramo del repository, non è possibile recuperare tali commit a meno che non si disponga del commit completo. IDs

 Note

Le istruzioni in questo argomento non possono essere utilizzate per eliminare un ramo predefinito di un repository. Se desideri eliminare il ramo predefinito, devi creare un ramo, impostarlo come ramo predefinito e quindi eliminare il ramo precedente. Per ulteriori informazioni, consultare [Creare un ramo](#) e [Modificare le impostazioni della filiale](#).

Argomenti

- [Eliminare un ramo \(console\)](#)
- [Elimina un ramo \(\)AWS CLI](#)
- [Eliminare un ramo \(Git\)](#)

Eliminare un ramo (console)

È possibile utilizzare la CodeCommit console per eliminare un ramo in un CodeCommit repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In Repositories (Repository) selezionare il nome del repository in cui si desidera eliminare un ramo.
3. Nel riquadro di navigazione selezionare Branches (Rami).
4. Individuare il nome del ramo da eliminare, scegliere Delete branch (Elimina ramo) e confermare la scelta.

Elimina un ramo ()AWS CLI

È possibile utilizzare il AWS CLI per eliminare un ramo in un CodeCommit repository, se quel ramo non è il ramo predefinito per il repository. Per ulteriori informazioni sull'installazione e l'utilizzo di AWS CLI, vedere. [Guida di riferimento alla riga di comando](#)

1. Dal terminale o dalla riga di comando, eseguire il comando delete-branch, specificando:
 - Il nome del CodeCommit repository in cui eliminare il ramo (con l'--repository-nameopzione).

 Tip

Per ottenere il nome del CodeCommit repository, esegui il comando [list-repositories](#).

- Il nome del ramo da eliminare (con l'opzione branch-name).

 Tip

[Per ottenere il nome del ramo, esegui il comando list-branches.](#)

2. Ad esempio, per eliminare un ramo denominato MyNewBranch in un CodeCommit repository denominato: MyDemoRepo

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

Questo comando restituisce informazioni sul ramo eliminato, tra cui il nome e l'ID commit completo del commit HEAD del ramo. Esempio:

```
"deletedBranch": {  
  "branchName": "MyNewBranch",  
  "commitId": "317f8570EXAMPLE"  
}
```

Eliminare un ramo (Git)

Segui questi passaggi per utilizzare Git da un repository locale per eliminare un ramo in un CodeCommit repository.

Questi passaggi sono scritti partendo dal presupposto che tu abbia già collegato il repository locale al repository. CodeCommit Per istruzioni, consulta [Connessione a un repository](#).

1. Per eliminare il ramo dal repository locale, esegui il git branch -D **branch-name** comando where **branch-name** è il nome del ramo che desideri eliminare.

 Tip

Per ottenere un elenco di nomi di rami, eseguire git branch --all.

Ad esempio, per eliminare un ramo nel repository locale denominato: MyNewBranch

```
git branch -D MyNewBranch
```

2. Per eliminare il ramo dal CodeCommit repository, esegui il git push **remote-name** --delete **branch-name** comando dove **remote-name** è il soprannome che il repository locale usa per il CodeCommit repository ed **branch-name** è il nome del ramo che desideri eliminare dal repository. CodeCommit

 Tip

Per ottenere un elenco dei nomi dei CodeCommit repository e relativi URLs, esegui il comando. git remote -v

Ad esempio, per eliminare un ramo denominato MyNewBranch nel CodeCommit repository denominato: origin

```
git push origin --delete MyNewBranch
```

 Tip

Questo comando non elimina un ramo impostato come predefinito.

Per altre opzioni, consulta la documentazione di Git.

Utilizzo delle preferenze dell'utente

Puoi utilizzare la AWS CodeCommit console per configurare alcune impostazioni predefinite. Ad esempio, puoi modificare le preferenze relative alla visualizzazione delle modifiche al codice, inline o in una vista combinata. Quando apporti una modifica a una di queste impostazioni, la AWS CodeCommit console imposta un cookie nel browser che memorizza e applica le tue scelte ogni volta che utilizzi la console. Queste preferenze vengono applicate a tutti i repository in tutte le regioni ogni volta che si accede alla AWS CodeCommit console utilizzando quel browser. Le preferenze non sono specifiche del repository o della regione. Non hanno alcun effetto sulle interazioni dell'utente con l'AWS CLI AWS CodeCommit API o altri servizi con AWS CodeCommit cui interagisce.

Note

I cookie delle preferenze utente sono specifici del browser. Se cancelli i cookie dal browser, le tue preferenze vengono cancellate. Analogamente, se usi un browser diverso per accedere a un repository, il browser non ha accesso ai cookie dell'altro browser. Le preferenze non vengono mantenute.

Le preferenze utente includono:

- Quando visualizzare le modifiche di codice, se utilizzare la vista Unified (Unificata) o Split (Doppia) e se mostrare o nascondere le modifiche degli spazi.
- Quando visualizzare, modificare o creare il codice, se utilizzare uno sfondo chiaro o scuro nella finestra dell'editor del codice.

La pagina per impostare le preferenze non è una sola. Al contrario, ogni volta che si modifica una preferenza nella console, ad esempio per visualizzare le modifiche al codice, la modifica viene salvata e applicata ogni volta che è necessario.

Migrare a AWS CodeCommit

Puoi migrare un repository Git in un CodeCommit repository in diversi modi: clonandolo, eseguendo il mirroring, migrando tutti o solo alcuni dei rami e così via. Puoi anche migrare ad CodeCommit il contenuto locale senza versione presente sul tuo computer.

I seguenti argomenti illustrano alcuni dei modi in cui è possibile eseguire la migrazione di un repository. La procedura potrebbe variare a seconda del tipo, dello stile o della complessità del repository e delle decisioni che hai preso in merito a cosa intendi migrare e con quali modalità. Per i repository di dimensioni molto grandi, puoi valutare di eseguire la [migrazione incrementale](#).

Note

Puoi migrare CodeCommit da altri sistemi di controllo delle versioni, come Perforce, Subversion o TFS, ma devi prima migrare a Git.

Per altre opzioni, consulta la documentazione di Git.

In alternativa, puoi vedere le informazioni sulla [migrazione a Git](#) nel manuale Pro Git di Scott Chacon e Ben Straub.

Argomenti

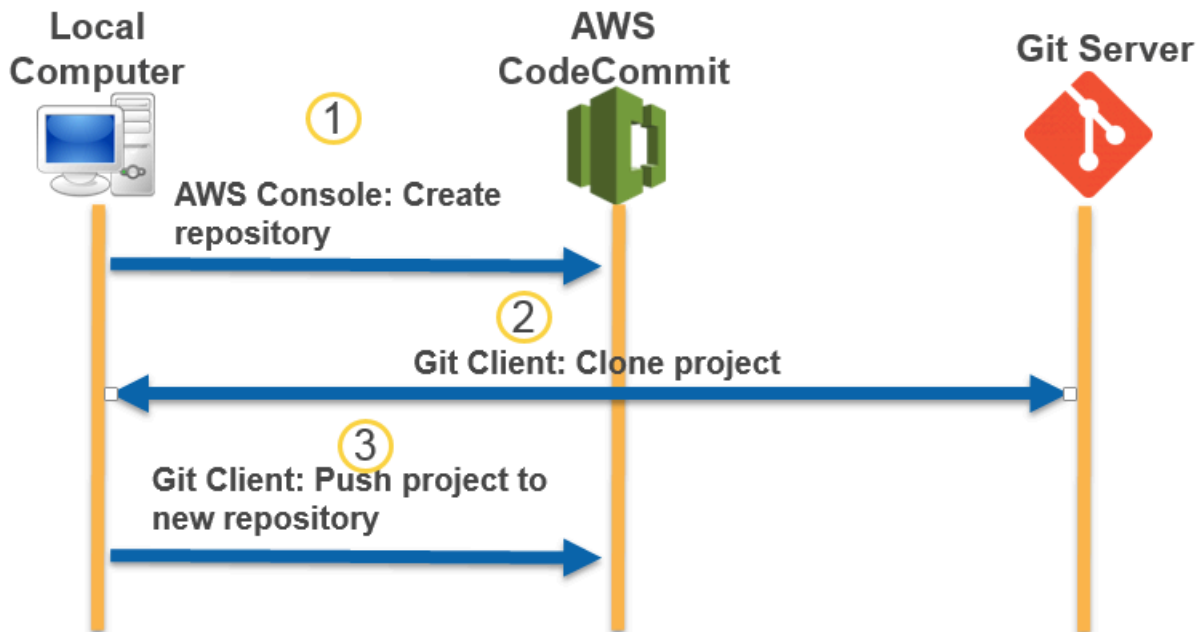
- [Migrare un repository Git su AWS CodeCommit](#)
- [Esegui la migrazione di contenuti locali o senza versioni su AWS CodeCommit](#)
- [Migrare un repository in modo incrementale](#)

Migrare un repository Git su AWS CodeCommit

È possibile migrare un repository Git esistente in un CodeCommit repository. Le procedure descritte in questo argomento illustrano il processo di migrazione in CodeCommit di un progetto ospitato in un altro repository Git. Per eseguire il processo:

- Completa la configurazione iniziale richiesta per. CodeCommit
- Crea un CodeCommit repository.
- Clona il repository e invialo a. CodeCommit
- Visualizza i file nel repository. CodeCommit

- Condividi il CodeCommit repository con il tuo team.



Argomenti

- [Fase 0: Configurazione richiesta per l'accesso a CodeCommit](#)
- [CodeCommit Fase 1: Creare un repository](#)
- [Passaggio 2: clonare il repository e inviarlo al repository CodeCommit](#)
- [Passaggio 3: Visualizza i file in CodeCommit](#)
- [Fase 4: Condividere il repository CodeCommit](#)

Fase 0: Configurazione richiesta per l'accesso a CodeCommit

Prima di poter migrare un repository verso CodeCommit, è necessario creare e configurare un utente IAM per l'accesso a CodeCommit e configurare il computer locale. È inoltre necessario installare AWS CLI per gestire CodeCommit. Sebbene sia possibile eseguire la maggior parte delle CodeCommit attività senza di essa, AWS CLI offre flessibilità quando si lavora con Git dalla riga di comando o dal terminale.

Se sei già configurato per CodeCommit, puoi passare direttamente a [CodeCommit Fase 1: Creare un repository](#).

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

 Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la politica da allegare, scegli Avanti: revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Per installare e configurare AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con `git-remote-codecommit`

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o esegui l'aggiornamento AWS CLI all'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando.

Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il `configure` comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS dove si trova il repository, ad esempio. `us-east-2` Quando viene richiesto il formato di output predefinito, specificare `json`. Ad esempio, se stai configurando un profilo per un utente 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*

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Quindi, installare Git.

- Per Linux, macOS o Unix:

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit
Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

- Per Windows:

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti Web come [Git per Windows](#). Se usi questo link per installare Git, puoi accettare tutte le impostazioni predefinite di installazione tranne le seguenti:

- Quando richiesto durante il passaggio Adjusting your PATH environment, scegli l'opzione per usare Git dalla riga di comando.
- (Facoltativo) Se intendi utilizzare HTTPS con l'helper per le credenziali incluso in AWS CLI invece di configurare le credenziali Git CodeCommit per, nella pagina Configurazione delle opzioni extra, assicurati che l'opzione Enable Git Credential Manager sia deselezionata. Git Credential Manager è compatibile solo CodeCommit se gli utenti IAM configurano le credenziali Git. Per ulteriori informazioni, consultare [Per utenti HTTPS che utilizzano credenziali Git](#) e [Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository \(403\)](#).

 Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit
Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

CodeCommit supporta l'autenticazione HTTPS e SSH. Per completare la configurazione, è necessario configurare le credenziali Git per CodeCommit (HTTPS, consigliato per la maggior parte degli utenti), una coppia di chiavi SSH da utilizzare per l'accesso CodeCommit (SSH), git-remote-codecommit (consigliato per gli utenti che utilizzano l'accesso federato) o l'helper per le credenziali incluso in (HTTPS). AWS CLI

- Per le credenziali Git su tutti i sistemi operativi supportati, vedi [Passaggio 3: Creare credenziali Git per le connessioni HTTPS a CodeCommit](#).
- Per SSH su Linux, macOS o Unix, vedi. [SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit](#)

- Per SSH su Windows, vedi [Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit](#).
- Per git-remote-codecommit, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).
- Per l'helper per le credenziali su Linux, macOS o Unix, consulta [Configurare l'helper delle credenziali \(Linux, macOS o Unix\)](#).
- Per l'assistente credenziali su Windows, vedi l'argomento relativo alla [configurazione dell'assistente credenziali \(Windows\)](#).

CodeCommit Fase 1: Creare un repository

In questa sezione, usi la CodeCommit console per creare il CodeCommit repository che utilizzerai per il resto di questo tutorial. Per usare il AWS CLI per creare il repository, vedi. [Crea un repository \(AWS CLI\)](#)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove vuoi creare il repository. Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository), scegli Create repository (Crea repository).
4. Nella pagina Create repository (Crea repository) immettere un nome per il repository in Repository name (Nome repository).

Note

I nomi dei repository distinguono tra maiuscole e minuscole. Il nome deve essere univoco nel Regione AWS tuo account Amazon Web Services.

5. (Facoltativo) In Description (Descrizione), immettere una descrizione per il repository. Ciò può consentire agli utenti di identificare lo scopo del repository.

Note

Il campo della descrizione mostra Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza GetRepository o BatchGetRepositories APIs e prevedi di visualizzare il campo di descrizione del repository in un browser web, consulta l'CodeCommit API Reference.](#)

6. (Facoltativo) Scegliete Aggiungi tag per aggiungere uno o più tag del repository (un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse) al repository. Per ulteriori informazioni, consulta [Taggare i repository in AWS CodeCommit](#).
7. (Facoltativo) Espandi la configurazione aggiuntiva per specificare se utilizzare la chiave predefinita Chiave gestita da AWS o la chiave gestita dal cliente per crittografare e decrittografare i dati in questo repository. Se scegli di utilizzare la tua chiave gestita dal cliente, devi assicurarti che sia disponibile nel luogo in Regione AWS cui stai creando il repository e che la chiave sia attiva. Per ulteriori informazioni, consulta [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#).
8. (Facoltativo) Seleziona Abilita Amazon CodeGuru Reviewer per Java e Python se questo repository contiene codice Java o Python e desideri che CodeGuru Reviewer lo analizzi. CodeGuru Reviewer utilizza diversi modelli di machine learning per trovare difetti nel codice e suggerire miglioramenti e correzioni nelle pull request. Per ulteriori informazioni, consulta la [Amazon CodeGuru Reviewer User Guide](#).
9. Scegli Create (Crea).

[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 volta creato, il repository viene visualizzato nell'elenco in [Repositories \(Repository\)](#). Nella colonna URL scegliere l'icona di copia, quindi scegliere il protocollo (HTTPS o SSH) da utilizzare per connettersi ad CodeCommit. Copia l'URL.

Ad esempio, se hai assegnato un nome al tuo repository *MyClonedRepository* e utilizzi credenziali Git con HTTPS nella regione Stati Uniti orientali (Ohio), l'URL è simile al seguente:

```
https://git-codecommit.us-east-2.amazonaws.com/MyClonedRepository
```

L'URL sarà necessario in seguito in [Passaggio 2: clonare il repository e inviarlo al repository CodeCommit](#).

Passaggio 2: clonare il repository e inviarlo al repository CodeCommit

In questa sezione, clonerai un repository Git nel computer locale, creando quello che viene definito un repository locale. Quindi trasferisci il contenuto del repository locale al CodeCommit repository creato in precedenza.

1. Dal terminale o dal prompt dei comandi del computer locale, esegui il git clone comando con l'--mirror opzione di clonare una semplice copia del repository remoto in una nuova cartella denominata. *aws-codecommit-demo* Questo è un repository base che serve esclusivamente per la migrazione. Non è il repository locale in cui interagire con il repository migrato. CodeCommit Puoi crearlo in un secondo momento, una volta completata la migrazione a. CodeCommit

L'esempio seguente clona un'applicazione demo ospitata su GitHub (<https://github.com/aws-labs/aws-demo-php-simple-app.git>) in un repository locale in una directory denominata. *aws-codecommit-demo*

```
git clone --mirror https://github.com/aws-labs/aws-demo-php-simple-app.git aws-codecommit-demo
```

2. Passare alla directory in cui è stato creato il clone.

```
cd aws-codecommit-demo
```

3. Eseguite il git push comando, specificando l'URL e il nome del CodeCommit repository di destinazione e l'opzione. --all (Si tratta dell'URL copiato in [CodeCommit Fase 1: Creare un repository](#)).

Ad esempio, se hai assegnato un nome al tuo repository *MyClonedRepository* e sei configurato per utilizzare HTTPS, eseguirai il comando seguente:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --all
```

Note

L'opzione --all esegue il push solo di tutti i rami per il repository e non di altri riferimenti, ad esempio i tag. Per eseguire il push dei tag, è necessario attendere il completamento del push iniziale, quindi eseguire un altro push utilizzando l'opzione --tags:

```
git push ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --tags
```

Per ulteriori informazioni, consulta [Git push](#) sul sito Web Git. Per informazioni sul push di repository di grandi dimensioni, ad esempio per eseguire il push di tutti i riferimenti in una

volta sola (ad esempio con l'opzione `--mirror`), vedi [Esegui la migrazione di un repository in modo incrementale](#).

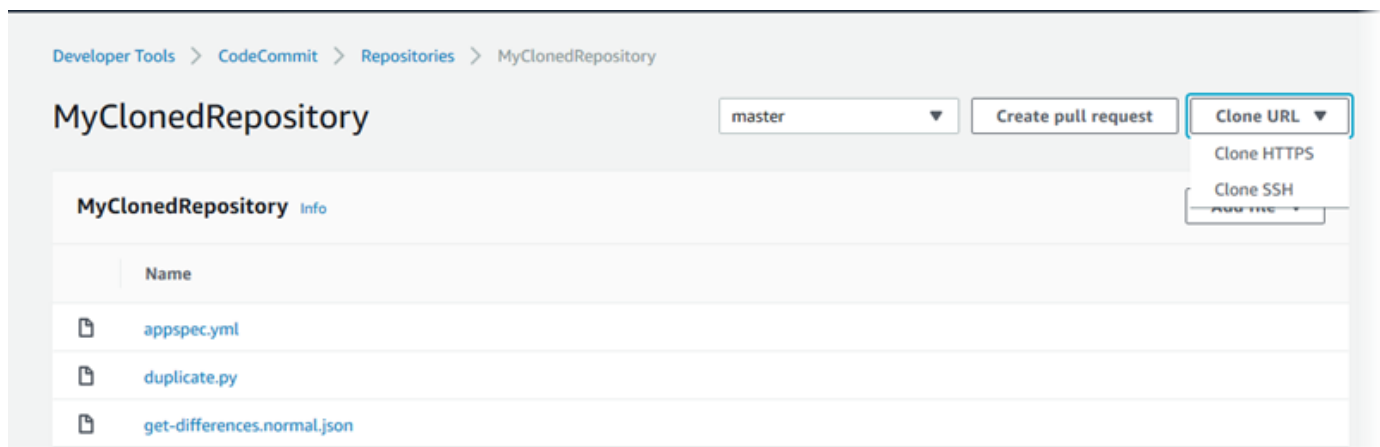
È possibile eliminare la `aws-codecommit-demo` cartella e il relativo contenuto dopo aver migrato il repository in CodeCommit. Per creare un repository locale con tutti i riferimenti corretti in cui lavorare con il repository CodeCommit, esegui il `git clone` comando senza l'opzione: `--mirror`

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

Passaggio 3: Visualizza i file in CodeCommit

Dopo aver inserito il contenuto della directory, è possibile utilizzare la CodeCommit console per visualizzare rapidamente tutti i file in quel repository.

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories, scegli il nome del repository (ad esempio,). `MyClonedRepository`
3. Visualizza i file nel repository per i rami, il clone URLs, le impostazioni e altro ancora.



Fase 4: Condividere il repository CodeCommit

Quando crei un repository in CodeCommit, vengono generati due endpoint: uno per le connessioni HTTPS e uno per le connessioni SSH. Entrambi offrono connessioni protette in rete. Gli utenti possono scegliere quale dei due protocolli utilizzare. Gli endpoint rimangono entrambi attivi, indipendentemente dal protocollo consigliato per i propri utenti. Prima di poter condividere il tuo repository con altri, devi creare politiche IAM che consentano ad altri utenti di accedere al tuo repository. Fornire queste istruzioni di accesso ai propri utenti.

Creazione di una policy gestita dal cliente per il repository

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nell'area di navigazione Dashboard (Pannello di controllo) scegliere Policies (Policy) e quindi Crea Policy (Crea policy).
3. Nella pagina Crea policy, scegli Importa policy gestita.
4. Nella pagina Importa policy gestite, in Filtra policy, inserisci **AWSCodeCommitPowerUser**. Scegli il pulsante accanto al nome della politica, quindi scegli Importa.
5. Nella pagina Create policy (Crea policy), scegli JSON. Sostituisci la parte «*» della Resource riga relativa CodeCommit alle azioni con l'Amazon Resource Name (ARN) del CodeCommit repository, come illustrato di seguito:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

Tip

Per trovare l'ARN per il CodeCommit repository, vai alla CodeCommit console, scegli il nome del repository dall'elenco, quindi scegli Impostazioni. Per ulteriori informazioni, consulta [Visualizza i dettagli del repository](#).

Per fare in modo che questa policy venga applicata a più repository, aggiungere ogni repository come una risorsa, specificandone l'ARN. Includere una virgola tra ciascuna dichiarazione di risorsa, come nell'esempio seguente:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Quando hai finito di modificare, scegli Review policy.

6. Nella pagina Rivedi politica, in Nome, inserisci un nuovo nome per la politica (ad esempio, **AWSCodeCommitPowerUser-MyDemoRepo**). Facoltativamente, fornisci una descrizione per questa politica.

7. Scegli Crea policy.

Per gestire l'accesso al tuo repository, crea un gruppo IAM per i relativi utenti, aggiungi gli utenti IAM a quel gruppo e quindi allega la policy gestita dai clienti che hai creato nel passaggio precedente. Allega qualsiasi altra policy richiesta per l'accesso, ad esempio IAMUser SSHKeys o IAMSelfManageServiceSpecificCredentials.

1. Accedi a Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nell'area di navigazione Dashboard (Pannello di controllo), scegliere Groups (Gruppi) e quindi Create New Group (Crea nuovo gruppo).
3. Nella pagina Imposta nome gruppo, in Nome gruppo, inserisci un nome per il gruppo (ad esempio, *MyDemoRepoGroup*), quindi scegli Passaggio successivo. Valutare la possibilità di includere il nome del repository nel nome del gruppo.

Note

Questo nome deve essere univoco per un account Amazon Web Services.

4. Seleziona la casella accanto alla politica gestita dai clienti che hai creato nella sezione precedente (ad esempio, AWSCodeCommitPowerUser-MyDemoRepo).
5. Nella pagina Review (Verifica), scegliere Create Group (Crea gruppo). IAM crea questo gruppo con le politiche specificate già allegate. Il gruppo viene visualizzato nell'elenco dei gruppi associati al tuo account Amazon Web Services.
6. Selezionare il gruppo dall'elenco.
7. Nella pagina di riepilogo del gruppo, scegliere la scheda Users (Utenti) e quindi scegliere Add Users to Group (Aggiungi utenti al gruppo). Nell'elenco che mostra tutti gli utenti associati al tuo account Amazon Web Services, seleziona le caselle accanto agli utenti a cui desideri consentire l'accesso al CodeCommit repository, quindi scegli Aggiungi utenti.

Tip

È possibile utilizzare la casella di ricerca per trovare rapidamente gli utenti per nome.

8. Dopo aver aggiunto gli utenti, chiudi la console IAM.

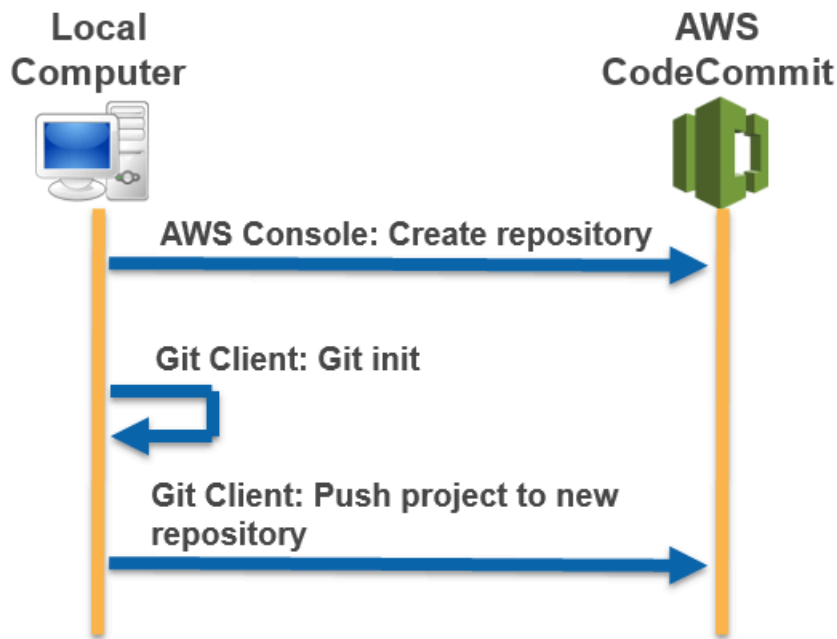
Dopo aver creato un utente IAM a cui accedere CodeCommit utilizzando il gruppo di policy e le policy che hai configurato, invia a quell'utente le informazioni necessarie per connettersi al repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository) scegliere il repository che si desidera eliminare.
4. In Clone URL (URL clone), scegliere il protocollo che deve essere utilizzato dagli utenti. L'URL clone viene copiato per il protocollo di connessione scelto.
5. Invia ai tuoi utenti l'URL clone insieme a qualsiasi altra istruzione, come l'installazione AWS CLI, la configurazione di un profilo o l'installazione di Git. Assicurarsi di includere le informazioni di configurazione per il protocollo di connessione (ad esempio HTTPS).

Esegui la migrazione di contenuti locali o senza versioni su AWS CodeCommit

Le procedure riportate in questo argomento mostrano come migrare un progetto esistente o un contenuto locale sul computer in un repository. CodeCommit Per eseguire il processo:

- Completare la configurazione iniziale richiesta per. CodeCommit
- Crea un CodeCommit repository.
- Posiziona una cartella locale sotto il controllo della versione Git e invia il contenuto di quella cartella al CodeCommit repository.
- Visualizza i file nel CodeCommit repository.
- Condividi il CodeCommit repository con il tuo team.



Argomenti

- [Fase 0: Configurazione richiesta per l'accesso a CodeCommit](#)
- [CodeCommit Fase 1: Creare un repository](#)
- [Fase 2: Migrazione dei contenuti locali nel repository CodeCommit](#)
- [Passaggio 3: Visualizza i file in CodeCommit](#)
- [Passaggio 4: Condividi il repository CodeCommit](#)

Fase 0: Configurazione richiesta per l'accesso a CodeCommit

Prima di poter migrare i contenuti locali verso CodeCommit, è necessario creare e configurare un utente IAM per l'accesso CodeCommit e configurare il computer locale per l'accesso. È inoltre necessario installare AWS CLI per gestire CodeCommit. Sebbene sia possibile eseguire la maggior parte delle CodeCommit attività senza di essa, AWS CLI offre flessibilità quando si lavora con Git.

Se sei già configurato per CodeCommit, puoi passare direttamente a [CodeCommit Fase 1: Creare un repository](#).

Per creare e configurare un utente IAM per l'accesso CodeCommit

1. Crea un account Amazon Web Services accedendo a <http://aws.amazon.com> e scegliendo Registrati.
2. Crea un utente IAM o usane uno esistente nel tuo account Amazon Web Services. Assicurati di avere un ID chiave di accesso e una chiave di accesso segreta associati a quell'utente IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo account Amazon Web Services](#).

 Note

CodeCommit richiede AWS Key Management Service. Se utilizzi un utente IAM esistente, assicurati che non vi siano policy associate all'utente che neghino espressamente le AWS KMS azioni richieste da CodeCommit. Per ulteriori informazioni, consulta [AWS KMS e crittografia](#).

3. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
4. Nella console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli l'utente IAM che desideri configurare per CodeCommit l'accesso.
5. Nella scheda Permissions (Autorizzazioni), scegliere Add permissions (Aggiungi autorizzazioni).
6. In Grant permissions (Concedi autorizzazioni), scegliere Attach existing policies directly (Collega direttamente le policy esistenti).
7. Dall'elenco delle politiche, seleziona AWSCodeCommitPowerUser o un'altra politica gestita per CodeCommit l'accesso. Per ulteriori informazioni, consulta [AWS politiche gestite per CodeCommit](#).

Dopo aver selezionato la policy da allegare, scegli Avanti: Revisione per esaminare l'elenco delle politiche da allegare all'utente IAM. Se l'elenco è corretto, scegliere Add permissions (Aggiungi autorizzazioni).

Per ulteriori informazioni sulle politiche CodeCommit gestite e sulla condivisione dell'accesso ai repository con altri gruppi e utenti, consulta [Condividi un repository](#) e [Autenticazione e controllo degli accessi per AWS CodeCommit](#).

Per installare e configurare AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con `git-remote-codecommit`

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o esegui l'aggiornamento AWS CLI all'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando.

Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il `configure` comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS dove si trova il repository, ad esempio. `us-east-2` Quando viene richiesto il formato di output predefinito, specificare `json`. Ad esempio, se stai configurando un profilo per un utente 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*

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Quindi, installare Git.

- Per Linux, macOS o Unix:

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit
Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

- Per Windows:

Per lavorare con file, commit e altre informazioni nei CodeCommit repository, devi installare Git sul tuo computer locale. CodeCommit supporta le versioni Git 1.7.9 e successive. La versione Git 2.28 supporta la configurazione del nome del ramo per i commit iniziali. Si consiglia di utilizzare una versione recente di Git.

Per installare Git, consigliamo siti Web come [Git per Windows](#). Se usi questo link per installare Git, puoi accettare tutte le impostazioni predefinite di installazione tranne le seguenti:

- Quando richiesto durante il passaggio Adjusting your PATH environment, scegli l'opzione per usare Git dalla riga di comando.
- (Facoltativo) Se intendi utilizzare HTTPS con l'helper per le credenziali incluso in AWS CLI invece di configurare le credenziali Git CodeCommit per, nella pagina Configurazione delle opzioni extra, assicurati che l'opzione Enable Git Credential Manager sia deselezionata. Git Credential Manager è compatibile solo CodeCommit se gli utenti IAM configurano le credenziali Git. Per ulteriori informazioni, consultare [Per utenti HTTPS che utilizzano credenziali Git](#) e [Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository \(403\)](#).

 Note

Git è una piattaforma in evoluzione e regolarmente aggiornata. Occasionalmente, una modifica delle funzionalità potrebbe influire sul modo in cui funziona. CodeCommit
Se riscontri problemi con una versione specifica di Git and CodeCommit, consulta le informazioni in [Risoluzione dei problemi](#).

CodeCommit supporta l'autenticazione HTTPS e SSH. Per completare la configurazione, è necessario configurare le credenziali Git per CodeCommit (HTTPS, consigliato per la maggior parte degli utenti), una coppia di chiavi SSH (SSH) da utilizzare per l'accesso CodeCommit git-remote-codecommit (consigliato per gli utenti che utilizzano l'accesso federato) o l'helper per le credenziali incluso in. AWS CLI

- Per le credenziali Git su tutti i sistemi operativi supportati, vedi [Passaggio 3: Creare credenziali Git per le connessioni HTTPS a CodeCommit](#).
- Per SSH su Linux, macOS o Unix, vedi. [SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit](#)

- Per SSH su Windows, vedi [Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit](#).
- Per git-remote-codecommit, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).
- Per l'helper per le credenziali su Linux, macOS o Unix, consulta [Configurare l'helper delle credenziali \(Linux, macOS o Unix\)](#).
- Per l'assistente credenziali su Windows, vedi l'argomento relativo alla [configurazione dell'assistente credenziali \(Windows\)](#).

CodeCommit Fase 1: Creare un repository

In questa sezione, usi la CodeCommit console per creare il CodeCommit repository che utilizzerai per il resto di questo tutorial. Per usare il AWS CLI per creare il repository, vedi. [Crea un repository \(AWS CLI\)](#)

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove vuoi creare il repository. Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository), scegli Create repository (Crea repository).
4. Nella pagina Create repository (Crea repository) immettere un nome per il repository in Repository name (Nome repository).

Note

I nomi dei repository distinguono tra maiuscole e minuscole. Il nome deve essere univoco nel Regione AWS tuo account Amazon Web Services.

5. (Facoltativo) In Description (Descrizione), immettere una descrizione per il repository. Ciò può consentire agli utenti di identificare lo scopo del repository.

Note

Il campo della descrizione mostra Markdown nella console e accetta tutti i caratteri HTML e i caratteri Unicode validi. [Se sei uno sviluppatore di applicazioni che utilizza GetRepository o BatchGetRepositories APIs e prevedi di visualizzare il campo di descrizione del repository in un browser web, consulta l'CodeCommit API Reference.](#)

6. (Facoltativo) Scegliete Aggiungi tag per aggiungere uno o più tag del repository (un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse) al repository. Per ulteriori informazioni, consulta [Taggare i repository in AWS CodeCommit](#).
7. (Facoltativo) Espandi la configurazione aggiuntiva per specificare se utilizzare la chiave predefinita Chiave gestita da AWS o la chiave gestita dal cliente per crittografare e decrittografare i dati in questo repository. Se scegli di utilizzare la tua chiave gestita dal cliente, devi assicurarti che sia disponibile nel luogo in Regione AWS cui stai creando il repository e che la chiave sia attiva. Per ulteriori informazioni, consulta [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#).
8. (Facoltativo) Seleziona Abilita Amazon CodeGuru Reviewer per Java e Python se questo repository contiene codice Java o Python e desideri che CodeGuru Reviewer lo analizzi. CodeGuru Reviewer utilizza diversi modelli di machine learning per trovare difetti nel codice e suggerire miglioramenti e correzioni nelle pull request. Per ulteriori informazioni, consulta la [Amazon CodeGuru Reviewer User Guide](#).
9. Scegli Create (Crea).

Una volta creato, il repository viene visualizzato nell'elenco in Repositories (Repository). Nella colonna URL scegliere l'icona di copia, quindi scegliere il protocollo (HTTPS o SSH) da utilizzare per connettersi ad CodeCommit. Copia l'URL.

Ad esempio, se hai dato un nome al tuo repository *MyFirstRepo* e utilizzi HTTPS, l'URL sarà simile al seguente:

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

L'URL sarà necessario in seguito in [Fase 2: Migrazione dei contenuti locali nel repository CodeCommit](#).

Fase 2: Migrazione dei contenuti locali nel repository CodeCommit

Ora che hai un CodeCommit repository, puoi scegliere una directory sul tuo computer locale da convertire in un repository Git locale. Il comando `git init` può essere utilizzato per convertire il contenuto senza versione esistente in un repository Git o, se ancora non sono disponibili file o contenuto, per inizializzare un nuovo repository vuoto.

1. Dal terminale o dalla riga di comando sul computer locale passare alla directory da usare come origine per il repository.

2. Esegui il seguente comando per configurare Git in modo che utilizzi un ramo predefinito denominato **main**:

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

Puoi anche eseguire questo comando per impostare il nome del ramo predefinito su tutti **main** i repository appena creati:

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

3. Eseguire il comando `git init` per inizializzare il controllo di versione Git nella directory. In questo modo si crea una sottodirectory `.git` nella radice della directory che consente il monitoraggio del controllo di versione. La cartella `.git` contiene anche tutti i metadati necessari per il repository.

```
git init
```

4. Controllate lo stato della directory inizializzata eseguendo il seguente comando:

```
git status
```

Aggiungere i file desiderati al controllo di versione. In questo tutorial eseguiremo il comando `git add` con l'identificatore `.` per aggiungere tutti i file in questa directory. Per altre opzioni, consulta la documentazione di Git.

```
git add .
```

5. Creare un commit per i file aggiunti con un messaggio di commit.

```
git commit -m "Initial commit"
```

6. Esegui il `git push` comando, specificando l'URL e il nome del CodeCommit repository di destinazione e l'opzione. `--all` (Si tratta dell'URL copiato in [CodeCommit Fase 1: Creare un repository](#)).

Ad esempio, se hai assegnato un nome al tuo repository *MyFirstRepo* e sei configurato per utilizzare HTTPS, eseguirai il comando seguente:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyFirstRepo --all
```

Passaggio 3: Visualizza i file in CodeCommit

Dopo aver inserito il contenuto della directory, è possibile utilizzare la CodeCommit console per visualizzare rapidamente tutti i file del repository.

1. [Apri la CodeCommit console su codecommit/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. In Repositories, scegli il nome del repository (ad esempio, *MyFirstRepository*) dall'elenco.
3. Visualizza i file nel repository per i rami, il clone URLs, le impostazioni e altro.

Passaggio 4: Condividi il repository CodeCommit

Quando crei un repository in CodeCommit, vengono generati due endpoint: uno per le connessioni HTTPS e uno per le connessioni SSH. Entrambi offrono connessioni protette in rete. Gli utenti possono scegliere quale dei due protocolli utilizzare. Gli endpoint rimangono entrambi attivi, indipendentemente dal protocollo consigliato per i propri utenti. Prima di poter condividere il tuo repository con altri, devi creare politiche IAM che consentano ad altri utenti di accedere al tuo repository. Fornire queste istruzioni di accesso ai propri utenti.

Creazione di una policy gestita dal cliente per il repository

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nell'area di navigazione Dashboard (Pannello di controllo) scegliere Policies (Policy) e quindi Crea Policy (Crea policy).
3. Nella pagina Crea policy, scegli Importa policy gestita.
4. Nella pagina Importa policy gestite, in Filtra policy, inserisci **AWSCodeCommitPowerUser**. Scegli il pulsante accanto al nome della politica, quindi scegli Importa.
5. Nella pagina Create policy (Crea policy), scegli JSON. Sostituisci la parte «*» della Resource riga relativa CodeCommit alle azioni con l'Amazon Resource Name (ARN) del CodeCommit repository, come illustrato di seguito:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
]
```

 Tip

Per trovare l'ARN per il CodeCommit repository, vai alla CodeCommit console, scegli il nome del repository dall'elenco, quindi scegli Impostazioni. Per ulteriori informazioni, consulta [Visualizza i dettagli del repository](#).

Per fare in modo che questa policy venga applicata a più repository, aggiungere ogni repository come una risorsa, specificandone l'ARN. Includere una virgola tra ciascuna dichiarazione di risorsa, come nell'esempio seguente:

```
"Resource": [  
  "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",  
  "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
]
```

Quando hai finito di modificare, scegli Review policy.

6. Nella pagina Rivedi politica, in Nome, inserisci un nuovo nome per la politica (ad esempio, *AWSCodeCommitPowerUser-MyDemoRepo*). Facoltativamente, fornisci una descrizione per questa politica.
7. Scegli Crea policy.

Per gestire l'accesso al tuo repository, crea un gruppo IAM per i relativi utenti, aggiungi gli utenti IAM a quel gruppo e quindi allega la policy gestita dai clienti che hai creato nel passaggio precedente. Collegare qualsiasi altra policy necessaria per l'accesso, ad esempio IAMSelfManageServiceSpecificCredentials o IAMUserSSHKeys.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nell'area di navigazione Dashboard (Pannello di controllo), scegliere Groups (Gruppi) e quindi Create New Group (Crea nuovo gruppo).
3. Nella pagina Imposta nome gruppo, in Nome gruppo, inserisci un nome per il gruppo (ad esempio, *MyDemoRepoGroup*), quindi scegli Passaggio successivo. Valutare la possibilità di includere il nome del repository nel nome del gruppo.

 Note

Questo nome deve essere univoco per un account Amazon Web Services.

4. Seleziona la casella accanto alla politica gestita dai clienti che hai creato nella sezione precedente (ad esempio, AWSCodeCommitPowerUser-MyDemoRepo).
5. Nella pagina Review (Verifica), scegliere Create Group (Crea gruppo). IAM crea questo gruppo con le politiche specificate già allegate. Il gruppo viene visualizzato nell'elenco dei gruppi associati al tuo account Amazon Web Services.
6. Selezionare il gruppo dall'elenco.
7. Nella pagina di riepilogo del gruppo, scegliere la scheda Users (Utenti) e quindi scegliere Add Users to Group (Aggiungi utenti al gruppo). Nell'elenco che mostra tutti gli utenti associati al tuo account Amazon Web Services, seleziona le caselle accanto agli utenti a cui desideri consentire l'accesso al CodeCommit repository, quindi scegli Aggiungi utenti.

 Tip

È possibile utilizzare la casella di ricerca per trovare rapidamente gli utenti per nome.

8. Dopo aver aggiunto gli utenti, chiudi la console IAM.

Dopo aver creato un utente IAM da utilizzare per l'accesso CodeCommit utilizzando il gruppo di policy e le policy che hai configurato, invia a quell'utente le informazioni necessarie per connettersi al repository.

1. Apri la CodeCommit console su <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Nel selettore della regione, scegli Regione AWS dove è stato creato il repository. I repository sono specifici di un. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git](#).
3. Nella pagina Repositories (Repository) scegliere il repository che si desidera eliminare.
4. In Clone URL (URL clone), scegliere il protocollo che deve essere utilizzato dagli utenti. L'URL clone viene copiato per il protocollo di connessione scelto.
5. Invia ai tuoi utenti l'URL del clone insieme a qualsiasi altra istruzione, come l'installazione AWS CLI, la configurazione di un profilo o l'installazione di Git. Assicurarsi di includere le informazioni di configurazione per il protocollo di connessione (ad esempio HTTPS).

Migrare un repository in modo incrementale

Quando esegui la migrazione a AWS CodeCommit, prendi in considerazione la possibilità di utilizzare il repository in incrementi o blocchi per ridurre le possibilità che un problema di rete intermittente o un peggioramento delle prestazioni della rete causino il fallimento dell'intera operazione. L'utilizzo dei push incrementali con uno script simile a quello incluso in questa pagina, consente di riavviare la migrazione ed eseguire solo il push dei commit che non sono riusciti nel tentativo precedente.

Le procedure in questo argomento illustrano come creare ed eseguire uno script che migra il repository in incrementi ed esegue un nuovo push dei soli incrementi non riusciti, finché la migrazione non è stata completata.

Queste istruzioni presuppongono che sia già stata completata la procedura in [Configurazione](#) e [Creazione di un repository](#).

Argomenti

- [Fase 0: determinare se effettuare la migrazione in modo incrementale](#)
- [Fase 1: Installare i prerequisiti e aggiungere il CodeCommit repository come telecomando](#)
- [Passaggio 2: creare lo script da utilizzare per la migrazione incrementale](#)
- [Passaggio 3: Esegui lo script ed esegui la migrazione incrementale a CodeCommit](#)
- [Appendice: script di esempio incremental-repo-migration.py](#)

Fase 0: determinare se effettuare la migrazione in modo incrementale

Prima di stabilire la dimensione globale del repository e decidere se effettuare una migrazione incrementale devi prendere in considerazione diversi fattori. Il più ovvio è la dimensione complessiva degli elementi inclusi nel repository. Anche un fattore come la cronologia cumulativa del repository può contribuire alla dimensione. Un repository con anni di storia e numerosi rami può essere molto grande, anche se i singoli asset non lo sono. Puoi adottare diverse strategie per rendere la migrazione di questi repository più semplice ed efficiente. Ad esempio, l'implementazione di una strategia di clone superficiale durante la clonazione di un repository caratterizzato da una lunga storia di sviluppo oppure la disattivazione della compressione delta per i file binari di grandi dimensioni. Puoi cercare le opzioni disponibili consultando la documentazione Git oppure puoi scegliere di impostare e configurare push incrementali per migrare il repository utilizzando lo script di esempio incluso nell'argomento `incremental-repo-migration.py`.

Puoi scegliere di configurare i push incrementali se si verifica una o più delle seguenti condizioni:

- Il repository che desideri migrare ha più di cinque anni di storia.
- La connessione a Internet è soggetta a interruzioni intermittenti, eliminazione di pacchetti, risposte lente o altre interruzioni durante il servizio.
- La dimensione complessiva del repository è superiore a 2 GB e hai intenzione di migrare l'intero repository.
- Il repository contiene elementi o file binari di grandi dimensioni che non si comprimono correttamente, ad esempio file di immagine di grandi dimensioni con più di cinque versioni tracciate.
- In precedenza hai tentato una migrazione verso CodeCommit e hai ricevuto un messaggio di «Errore interno del servizio».

Anche se nessuna delle condizioni appena indicate si verifica, puoi comunque scegliere di eseguire il push in modo incrementale.

Fase 1: Installare i prerequisiti e aggiungere il CodeCommit repository come telecomando

Puoi creare uno script personalizzato con prerequisiti specifici. Se utilizzi l'esempio incluso in questo argomento, devi:

- Installarne i prerequisiti.
- Clonare il repository sul tuo computer locale.
- Aggiungi il CodeCommit repository come remoto per il repository che desideri migrare.

Configura per eseguire `.py incremental-repo-migration`

1. Sul computer locale, installare Python 2.6 o versione successiva. Per ulteriori informazioni e per le versioni più recenti, vedere [il sito Web di Python](#).
2. Sullo stesso computer, installa GitPython, che è una libreria Python usata per interagire con i repository Git. Per ulteriori informazioni, consulta [la GitPython documentazione](#).
3. Utilizzare il comando `git clone --mirror` per clonare il repository da migrare sul computer locale. Dal terminale (Linux, macOS o Unix) o dal prompt dei comandi (Windows), usa il `git clone --mirror` comando per creare un repository locale per il repository, inclusa la directory in cui desideri creare il repository locale. Ad esempio, per clonare un repository Git denominato

MyMigrationRepo con un URL o in una directory denominata: *https://example.com/my-repo/my-repo*

```
git clone --mirror https://example.com/my-repo/MyMigrationRepo.git my-repo
```

Dovrebbe essere visualizzato un output simile al seguente, che indica che il repository è stato clonato in un repository locale essenziale denominato 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 le directory nel repository locale per il repository che hai appena clonato (ad esempio,). *my-repo* Da questa directory, utilizzare il comando `git remote add` **DefaultRemoteName** **RemoteRepositoryURL** per aggiungere il repository CodeCommit come repository remoto per il repository locale.

Note

Quando si esegue il push di repository di grandi dimensioni, è consigliabile utilizzare SSH anziché HTTPS. Quando si esegue il push di una modifica estesa, di un numero elevato di modifiche oppure di un repository di grandi dimensioni, le connessioni HTTPS con esecuzione prolungata vengono spesso terminate prematuramente a causa di problemi di rete o delle impostazioni del firewall. Per ulteriori informazioni sulla configurazione di SSH, consulta CodeCommit o. [Per connessioni SSH su Linux, macOS o Unix](#) [Per le connessioni SSH su Windows](#)

Ad esempio, usa il comando seguente per aggiungere l'endpoint SSH per un CodeCommit repository *MyDestinationRepo* denominato repository remoto per il nome remoto: `codecommit`

```
git remote add codecommit ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDestinationRepo
```

Tip

Poiché è un clone, il nome di default del remoto (`origin`) è già in uso. È quindi necessario utilizzare un altro nome di remoto. Anche se l'esempio utilizza `codecommit`, è possibile immettere qualsiasi nome. Specificare il comando `git remote show` per esaminare l'elenco dei remoti impostati per il repository locale.

5. Specificare il comando `git remote -v` per visualizzare le impostazioni di recupero e di push per il repository locale e confermarne la corretta impostazione. Per esempio:

```
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)
```

Tip

Se sono ancora visualizzate voci di recupero e push per un altro repository remoto, ad esempio le voci per `origin`, rimuoverle utilizzando il comando `git remote set-url --delete`.

Passaggio 2: creare lo script da utilizzare per la migrazione incrementale

Queste istruzioni presuppongono che si sta utilizzando lo script di esempio `incremental-repo-migration.py`.

1. Aprire un editor di testo e incollare il contenuto dello [script di esempio](#) in un documento vuoto.
2. Salvare il documento in una directory di documenti (non la directory di lavoro del repository locale) e denominarlo `incremental-repo-migration.py`. Verificare che la directory scelta sia configurata nell'ambiente locale o nelle variabili di percorso, in modo da poter eseguire lo script Python dalla riga di comando o sul terminale.

Passaggio 3: Esegui lo script ed esegui la migrazione incrementale a CodeCommit

Ora che hai creato `incremental-repo-migration.py` lo script, puoi utilizzarlo per migrare in modo incrementale un repository locale in un repository. CodeCommit Per impostazione predefinita, lo script esegue il push dei commit in batch di 1.000 commit e tenta di utilizzare le impostazioni Git per la directory da cui viene eseguito come impostazioni per il repository locale e quello remoto. È possibile utilizzare le opzioni incluse in `incremental-repo-migration.py` per configurare altre impostazioni, se necessario.

1. Dal terminale o dal prompt dei comandi, modificare le directory del repository locale che si desidera migrare.
2. Da tale directory, eseguire il comando seguente:

```
python incremental-repo-migration.py
```

3. Lo script viene eseguito e mostra l'avanzamento sul terminale o nel prompt dei comandi. La visualizzazione dell'avanzamento è lenta per alcuni repository di grandi dimensioni. Lo script viene interrotto se un singolo push ha esito negativo per tre volte. È quindi possibile eseguirlo di nuovo, partendo dal batch che ha restituito esito negativo. È possibile rieseguire lo script finché tutti i push non avranno esito positivo e la migrazione non sarà stata completata.

Tip

È possibile eseguire `incremental-repo-migration.py` da qualsiasi directory a condizione che vengano utilizzate le opzioni `-l` e `-r` per specificare le impostazioni locali e remote da utilizzare. Ad esempio, per utilizzare lo script da qualsiasi directory per migrare un repository locale situato in `/tmp/` verso un telecomando soprannominato: *my-repo codecommit*

```
python incremental-repo-migration.py -l "/tmp/my-repo" -r "codecommit"
```

È anche possibile utilizzare l'opzione `-b` per modificare la dimensione predefinita dei batch utilizzata per il push incrementale. Se, ad esempio, si sta regolarmente eseguendo il push di un repository con file binari di grandi dimensioni che cambiano spesso e vengono utilizzati da un percorso con larghezza di banda di rete limitata, è possibile utilizzare l'opzione `-b` per modificare la dimensione del batch in 500 anziché 1.000. Per esempio:

```
python incremental-repo-migration.py -b 500
```

Ciò consente di eseguire il push incrementale del repository locale in batch di 500 commit. Se si decide di modificare di nuovo la dimensione dei batch durante la migrazione del repository, ad esempio se si decide di ridurre la dimensione dei batch dopo un tentativo non riuscito, ricordarsi di utilizzare l'opzione `-c` per rimuovere i tag dei batch prima di ripristinare la dimensione dei batch con l'opzione `-b`:

```
python incremental-repo-migration.py -c  
python incremental-repo-migration.py -b 250
```

Important

Non utilizzare l'opzione `-c` se si desidera eseguire di nuovo lo script dopo un esito negativo. L'opzione `-c` rimuove i tag utilizzati per creare batch di commit. Utilizza l'opzione `-c` solo se si desidera modificare la dimensione dei batch e riavviare la procedura oppure se si sceglie di non utilizzare più lo script.

Appendice: script di esempio **incremental-repo-migration.py**

Per praticità, abbiamo sviluppato lo script Python di esempio `incremental-repo-migration.py` per eseguire il push incrementale di un repository. Questo script è un codice di esempio open source e viene fornito senza alcuna modifica.

```
# 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("\tObjects: %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
)
```

Sicurezza in AWS CodeCommit

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e te. Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS CodeCommit, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo CodeCommit. I seguenti argomenti mostrano come eseguire la configurazione CodeCommit per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere CodeCommit le tue risorse.

Argomenti

- [Protezione dei dati in AWS CodeCommit](#)
- [Identity and Access Management per AWS CodeCommit](#)
- [Resilienza in AWS CodeCommit](#)
- [Sicurezza dell'infrastruttura in AWS CodeCommit](#)

Protezione dei dati in AWS CodeCommit

In quanto servizio gestito, è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per

progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

CodeCommit i repository vengono crittografati automaticamente quando sono inattivi. Non è richiesta alcuna azione del cliente. CodeCommit crittografa anche i dati del repository in transito. È possibile utilizzare il protocollo HTTPS, il protocollo SSH o entrambi con CodeCommit i repository. Per ulteriori informazioni, consulta [Configurazione per AWS CodeCommit](#). Puoi anche configurare l'[accesso ai repository da più account](#). CodeCommit

Argomenti

- [AWS Key Management Service e crittografia per i AWS CodeCommit repository](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

AWS Key Management Service e crittografia per i AWS CodeCommit repository

I dati nei CodeCommit repository vengono crittografati in transito e a riposo. Quando i dati vengono inseriti in un CodeCommit repository (ad esempio, chiamando `git push`), CodeCommit crittografa i dati ricevuti man mano che vengono archiviati nel repository. Quando i dati vengono estratti da un CodeCommit repository (ad esempio, effettuando una chiamata `git pull`), li CodeCommit decripta e quindi li invia al chiamante. Ciò presuppone che l'utente IAM associato alla richiesta push o pull sia stato autenticato da AWS. I dati inviati o ricevuti vengono trasmessi mediante i protocolli di rete crittografati HTTPS o SSH.

Puoi utilizzare una chiave gestita da AWS o una chiave gestita dal cliente per crittografare e decrittografare i dati nel tuo repository. Per ulteriori informazioni sulle differenze tra le chiavi gestite dal cliente e Chiavi gestite da AWS, consulta [Customer managed keys](#) e [Chiavi gestite da AWS](#). Se non specifichi una chiave gestita dal cliente, CodeCommit utilizzerà una Chiave gestita da AWS per

crittografare e decrittografare i dati nel tuo repository. Questo Chiave gestita da AWS viene creato automaticamente per te nel tuo Account AWS. La prima volta che crei un CodeCommit repository in un nuovo Regione AWS account Amazon Web Services, se non specifichi una chiave gestita dal cliente, CodeCommit crea una Chiave gestita da AWS (la `aws/codecommit` chiave) nello stesso account Regione AWS in AWS Key Management Service (AWS KMS). Questa `aws/codecommit` chiave viene utilizzata solo da CodeCommit. È memorizzato nel tuo account Amazon Web Services. A seconda di ciò che specifichi, CodeCommit utilizza la chiave gestita dal cliente o Chiave gestita da AWS per crittografare e decrittografare i dati nel repository.

Important

CodeCommit esegue le seguenti AWS KMS azioni sulla AWS KMS chiave utilizzata per crittografare e decrittografare i dati in un repository. Se si utilizza un Chiave gestita da AWS, un utente non necessita di autorizzazioni esplicite per queste azioni, ma non deve disporre di politiche allegate che neghino tali azioni per la chiave. `aws/codecommit` Se utilizzi una chiave gestita dal cliente con un Account AWS ID impostato come criterio principale per quella chiave, queste autorizzazioni devono essere impostate esplicitamente su `allow`. In particolare, quando si crea il primo repository e si aggiornano le chiavi per il repository, non è necessario impostare nessuna delle seguenti autorizzazioni `deny` se si utilizza un Chiave gestita da AWS, e devono essere impostate su `allow` se si utilizza una chiave gestita dal cliente con un criterio principale:

- `"kms:Encrypt"`
- `"kms:Decrypt"`
- `"kms:ReEncrypt"` (a seconda del contesto, ciò potrebbe richiedere, `kms:ReEncryptFrom`
`kms:ReEncryptTo`, o `kms:ReEncrypt*` non impostato per negare)
- `"kms:GenerateDataKey"`
- `"kms:GenerateDataKeyWithoutPlaintext"`
- `"kms:DescribeKey"`

Se si desidera utilizzare la propria chiave gestita dal cliente, la chiave deve essere disponibile nel luogo in Regione AWS cui si trova l'archivio. CodeCommit supporta l'utilizzo di chiavi gestite dal cliente sia singole che multiregionali. Sebbene siano supportati tutti i principali tipi di origine dei materiali, consigliamo di utilizzare l'opzione KMS predefinita. I clienti che utilizzano l'opzione External

Key Store potrebbero subire ritardi da parte del fornitore del negozio. Inoltre, CodeCommit presenta i seguenti requisiti per le chiavi gestite dal cliente:

- CodeCommit supporta solo l'utilizzo di chiavi simmetriche.
- Il tipo di utilizzo della chiave deve essere impostato su Encrypt and decrypt.

[Per ulteriori informazioni sulla creazione di chiavi gestite dal cliente, consulta Concetti e creazione di chiavi.](#)

Per visualizzare le informazioni sul file Chiave gestita da AWS generato da CodeCommit, procedi come segue:

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione del servizio, scegli. Chiavi gestite da AWS Assicurati di aver effettuato l'accesso al sito in Regione AWS cui desideri rivedere le chiavi.
4. Nell'elenco delle chiavi di crittografia, scegli Chiave gestita da AWS con l'alias aws/codecommit. Vengono visualizzate le informazioni di base su. Chiave di proprietà di AWS

Non è possibile modificarlo o eliminarlo Chiave gestita da AWS.

Come vengono utilizzati gli algoritmi di crittografia per crittografare i dati del repository

CodeCommit utilizza due approcci diversi per la crittografia dei dati. I singoli oggetti Git sotto 6 MB vengono crittografati utilizzando AES-GCM-256, che fornisce la convalida dell'integrità dei dati. Gli oggetti tra 6 MB e un massimo di 2 GB per un singolo blob vengono crittografati utilizzando AES-CBC-256. CodeCommit convalida sempre il contesto di crittografia.

Contesto di crittografia

Ogni servizio integrato con AWS KMS specifica un contesto di crittografia per le operazioni di crittografia e decrittografia. Il contesto di crittografia rappresenta informazioni autenticate supplementari utilizzate da AWS KMS per verificare l'integrità dei dati. Se specificato per l'operazione di crittografia, deve anche essere specificato nell'operazione di decrittografia. In caso contrario, la decrittografia non riesce. CodeCommit utilizza l'ID del CodeCommit repository per il contesto di

crittografia. È possibile utilizzare il `get-repository` comando o la CodeCommit console per trovare l'ID del repository. Cerca l'ID del CodeCommit repository nei AWS CloudTrail log per capire quali operazioni di crittografia sono state eseguite su quale chiave per crittografare o AWS KMS decrittografare i dati nel repository. CodeCommit

[Per ulteriori informazioni in merito, consulta la Guida per gli sviluppatori AWS KMS.AWS Key Management Service](#)

Connessione ai AWS CodeCommit repository con credenziali rotanti

Puoi consentire agli utenti di accedere ai tuoi AWS CodeCommit repository senza configurare gli utenti IAM per loro o utilizzare una chiave di accesso e una chiave segreta. Per assegnare le autorizzazioni a un'identità federata, crei un ruolo e definisci le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Create a role for a third-party identity provider \(federation\)](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center . Puoi anche configurare l'accesso basato sui ruoli per consentire agli utenti IAM di accedere ai CodeCommit repository in account Amazon Web Services separati (una tecnica nota come accesso tra account). Per una panoramica della configurazione dell'accesso tra account a un repository, consulta [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#).

Puoi anche configurare l'accesso per gli utenti che desiderano o devono effettuare l'autenticazione tramite metodi quali:

- Security Assertion Markup Language (SAML)
- Autenticazione a più fattori (MFA)
- Federazione
- Login with Amazon
- Amazon Cognito
- Facebook
- Google
- Provider di identità compatibile con OpenID Connect (OIDC)

 Note

Le seguenti informazioni si applicano solo all'uso `git-remote-codecommit` o all'aiuto delle AWS CLI credenziali per la connessione ai repository. CodeCommit Poiché l'approccio consigliato per l'accesso temporaneo o federato CodeCommit è l'impostazione `git-remote-codecommit`, questo argomento fornisce esempi di utilizzo di tale utilità. Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con `git-remote-codecommit`](#).

Non è possibile utilizzare credenziali SSH o Git e HTTPS per connettersi ai CodeCommit repository con credenziali di accesso rotanti o temporanee.

Non è necessario completare queste fasi se sono soddisfatti tutti i seguenti requisiti:

- Hai effettuato l'accesso a un' EC2 istanza Amazon.
- Stai utilizzando Git e HTTPS con l'helper delle AWS CLI credenziali per connetterti dall' EC2 istanza Amazon ai CodeCommit repository.
- L' EC2 istanza Amazon ha un profilo di istanza IAM allegato che contiene le autorizzazioni di accesso descritte in [Per connessioni HTTPS su Linux, macOS o Unix con l'helper delle credenziali AWS CLI](#) o [Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI](#).
- Hai installato e configurato l'helper per le credenziali Git sull' EC2 istanza Amazon, come descritto in [Per connessioni HTTPS su Linux, macOS o Unix con l'helper delle credenziali AWS CLI](#) o [Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI](#).

EC2 Le istanze Amazon che soddisfano i requisiti precedenti sono già configurate per comunicare credenziali di accesso temporanee CodeCommit a tuo nome.

 Note

Puoi configurare e utilizzare `git-remote-codecommit` su EC2 istanze Amazon.

Per consentire agli utenti l'accesso temporaneo ai tuoi CodeCommit repository, completa i seguenti passaggi.

Fase 1: completamento dei prerequisiti

Completa i passaggi di configurazione per fornire a un utente l'accesso ai tuoi CodeCommit repository utilizzando credenziali rotanti:

- Per l'accesso su più account, consulta [Procedura dettagliata: delega dell'accesso tra account Amazon Web Services utilizzando ruoli IAM](#) e [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#)
- Per SAML e federazione, consulta [Utilizzo del sistema di autenticazione dell'organizzazione per concedere l'accesso alle AWS risorse e Informazioni sulla federazione basata su SAML 2.0. AWS STS](#)
- Per l'MFA, consulta [Utilizzo di dispositivi Multi-Factor Authentication \(MFA\) AWS con e creazione di credenziali di sicurezza temporanee](#) per abilitare l'accesso per gli utenti IAM.
- Per Login with Amazon, Amazon Cognito, Facebook, Google o qualsiasi provider di identità compatibile con OIDC, consulta [Informazioni su Web Identity Federation. AWS STS](#)

Utilizza le informazioni in [Autenticazione e controllo degli accessi per AWS CodeCommit](#) per specificare le CodeCommit autorizzazioni che desideri concedere all'utente.

Fase 2: Ottenere il nome del ruolo o le credenziali di accesso

Se desideri che gli utenti accedano ai repository assumendo un ruolo, fornisci agli utenti l'Amazon Resource Name (ARN) del ruolo. In caso contrario, a seconda del modo in cui hai configurato l'accesso, l'utente può ottenere la rotazione delle credenziali in uno dei seguenti modi:

- Per l'accesso su più account, chiama il comando AWS CLI [assume-role](#) o chiama l'API. AWS STS [AssumeRole](#)
- Per SAML, chiama il AWS CLI [assume-role-with-saml](#) comando o l'API SAML. AWS STS [AssumeRoleWithSAML](#)
- Per la federazione, chiama AWS CLI [assume-role](#) o commands o o [get-federation-token](#). AWS STS [AssumeRoleGetFederationToken](#) APIs
- Per MFA, chiama il AWS CLI [get-session-token](#) comando o l' AWS STS [GetSessionToken](#) API.
- Per Login with Amazon, Amazon Cognito, Facebook, Google o qualsiasi provider di identità compatibile con OIDC, chiama il comando AWS CLI [assume-role-with-web-identity](#) o l'API. AWS STS [AssumeRoleWithWebIdentity](#)

Passaggio 3: installa e configura git-remote-codecommit AWS CLI

È necessario configurare il computer locale per utilizzare le credenziali di accesso installando [git-remote-codecommit](#) e configurando un profilo nell' AWS CLI.

1. Segui le istruzioni in [Configurazione](#) per configurare la AWS CLI. Utilizza il comando `aws configure` per configurare uno o più profili. Prendi in considerazione la creazione di un profilo denominato da utilizzare quando ti connetti ai CodeCommit repository utilizzando credenziali rotanti.
2. È possibile associare le credenziali al profilo AWS CLI denominato dell'utente in uno dei seguenti modi.
 - Se stai assumendo un ruolo per accedere CodeCommit, configura un profilo denominato con le informazioni necessarie per assumere quel ruolo. Ad esempio, se desideri assumere un ruolo denominato *CodeCommitAccess* nell'account Amazon Web Services 1111, puoi configurare un profilo predefinito da utilizzare quando lavori con altre AWS risorse e un profilo denominato da utilizzare quando assumi quel ruolo. I seguenti comandi creano un profilo denominato *CodeAccess* che assume un ruolo denominato *CodeCommitAccess*. Il nome utente *Maria_Garcia* è associato alla sessione e il profilo predefinito è impostato come origine delle relative AWS credenziali:

```
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
```

Per verificare le modifiche, visualizza o modifica manualmente il file `~/.aws/config` (per Linux) o il file `%UserProfile%.aws\config` (per Windows) e rivedi le informazioni sotto il profilo con nome. Il file ad esempio potrebbe essere simile al seguente.

```
[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
```

Dopo aver configurato il profilo con nome, è possibile clonare i repository CodeCommit con l'utilità `git-remote-codecommit` utilizzando il profilo con nome. Ad esempio, per clonare un repository con nome *MyDemoRepo*:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

- Se utilizzi la federazione delle identità web e OpenID Connect (OIDC), configura un profilo denominato che effettui la chiamata `AssumeRoleWithWebIdentity` API AWS Security Token Service (AWS STS) per tuo conto per aggiornare le credenziali temporanee. Utilizzate il `aws configure set` comando o modificate manualmente il `~/.aws/credentials` file (per Linux) o il `%UserProfile%.aws\credentials` file (per Windows) per aggiungere un profilo AWS CLI denominato con i valori di impostazione richiesti. Ad esempio, per creare un profilo che assuma il *CodeCommitAccess* ruolo e utilizzi un file di token di identità 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
```

Per ulteriori informazioni, consulta [Configurazione AWS Command Line Interface](#) e [utilizzo di un ruolo IAM nella Guida per l' AWS CLI](#)utente.AWS Command Line Interface

Fase 4: Accedere ai repository CodeCommit

Supponendo che l'utente abbia seguito le istruzioni per connettersi [Connessione a un repository](#) ai CodeCommit repository, l'utente utilizza quindi le funzionalità estese fornite da `git-remote-codecommit` Git per chiamare `git clone` e `git pull` clonare, inviare ed estrarre i CodeCommit repository a cui ha accesso. `git push` Ad esempio, per clonare un repository:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

I comandi Git `commit`, `push` e `pull` utilizzano la sintassi Git regolare.

Quando l'utente utilizza AWS CLI e specifica il profilo AWS CLI denominato associato alle credenziali di accesso rotanti, vengono restituiti i risultati relativi a quel profilo.

Identity and Access Management per AWS CodeCommit

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. CodeCommit IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Autenticazione e controllo degli accessi per AWS CodeCommit](#)
- [Come AWS CodeCommit funziona con IAM](#)
- [Policy di CodeCommit basate sulle risorse](#)
- [Autorizzazione basata su tag CodeCommit](#)
- [CodeCommit Ruoli IAM](#)
- [AWS CodeCommit esempi di politiche basate sull'identità](#)
- [Risoluzione dei problemi relativi AWS CodeCommit all'identità e all'accesso](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta [Risoluzione dei problemi relativi AWS CodeCommit all'identità e all'accesso](#))
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta [Come AWS CodeCommit funziona con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta [AWS CodeCommit esempi di politiche basate sull'identità](#))

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente di IAM.

Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida](#) per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica della lista di controllo degli accessi \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Autenticazione e controllo degli accessi per AWS CodeCommit

L'accesso a AWS CodeCommit richiede credenziali. Queste credenziali devono avere le autorizzazioni per accedere alle AWS risorse, come i CodeCommit repository, e al tuo utente IAM, che usi per gestire le tue credenziali Git o la chiave pubblica SSH che usi per creare connessioni Git. Le seguenti sezioni forniscono dettagli su come utilizzare [AWS Identity and Access Management \(IAM\)](#) e su come proteggere l'accesso CodeCommit alle risorse:

- [Autenticazione](#)
- [Controllo accessi](#)

Autenticazione

Poiché i CodeCommit repository sono basati su Git e supportano le funzionalità di base di Git, incluse le credenziali Git, ti consigliamo di utilizzare un utente IAM quando lavori con CodeCommit. Puoi accedere CodeCommit con altri tipi di identità, ma gli altri tipi di identità sono soggetti a limitazioni, come descritto di seguito.

Tipi di identità:

- Utente IAM: un [utente IAM](#) è un'identità all'interno del tuo account Amazon Web Services che dispone di autorizzazioni personalizzate specifiche. Ad esempio, un utente IAM può disporre delle autorizzazioni per creare e gestire le credenziali Git per l'accesso ai CodeCommit repository. Questo è il tipo di utente consigliato con cui lavorare. CodeCommit Puoi utilizzare un nome utente e una password IAM per accedere a AWS pagine Web sicure come [AWS Discussion Forums](#) o [Supporto AWS Center](#). [Console di gestione AWS](#)

Puoi generare credenziali Git o associare chiavi pubbliche SSH al tuo utente IAM, oppure puoi installare e configurare `git-remote-codecommit`. Questi sono i modi più semplici per configurare Git in modo che funzioni con i tuoi CodeCommit repository. Con [le credenziali Git](#), generi un nome utente e una password statici in IAM. Potrai quindi utilizzare queste credenziali per le connessioni HTTPS con Git e con qualsiasi strumento di terze parti che supporti l'autenticazione di nome utente e password Git. Con le connessioni SSH, crei file di chiave pubblici e privati sulla tua macchina locale che Git e Git CodeCommit utilizzano per l'autenticazione SSH. Associate la chiave pubblica al vostro utente IAM e memorizzate la chiave privata sul computer locale. [git-remote-codecommit](#) estende Git stesso e non richiede l'impostazione delle credenziali Git per l'utente.

Inoltre, puoi generare [chiavi di accesso](#) per ogni utente. [Utilizza le chiavi di accesso quando accedi ai AWS servizi a livello di codice, tramite uno dei AWS SDKs o utilizzando ilAWS Command Line Interface \(\).AWS CLI](#) L'SDK e gli strumenti CLI utilizzano le chiavi di accesso per firmare crittograficamente le tue richieste. Se non utilizzi gli AWS strumenti, devi firmare tu stesso le richieste. CodeCommit supporta Signature Version 4, un protocollo per l'autenticazione delle richieste API in entrata. Per ulteriori informazioni sulle richieste di autenticazione, consulta la pagina relativa al [processo di firma Signature Version 4](#) nella Riferimenti generali di AWS.

- Utente root dell'account Amazon Web Services: quando ti registri AWS, fornisci un indirizzo e-mail e una password associati al tuo account Amazon Web Services. Si tratta delle tue credenziali root, che forniscono accesso completo a tutte le risorse AWS . Alcune CodeCommit funzionalità non sono disponibili per gli utenti con account root. Inoltre, l'unico modo per usare Git con il tuo account root è installare e configurare git-remote-codecommit (consigliato) o configurare l'helper per AWS le credenziali, incluso in. AWS CLI Non puoi usare credenziali Git o coppie di chiavi SSH pubbliche-private con il tuo utente dell'account root. Per questi motivi, non è consigliabile utilizzare l'utente dell'account root per interagire con. CodeCommit

Important

Per ragioni di sicurezza, consigliamo di utilizzare le credenziali root solo per creare un utente amministratore, ovvero un utente IAM con autorizzazioni complete per il tuo account AWS . Potrai quindi utilizzare questo utente amministratore per creare altri utenti e ruoli IAM con autorizzazioni limitate. Per ulteriori informazioni consulta [Best practice IAM](#) e l'argomento relativo alla [creazione di un gruppo e un utente admin](#) nella Guida per l'utente IAM.

- IAM Identity Center e utenti in IAM Identity Center: AWS IAM Identity Center amplia le funzionalità di AWS Identity and Access Management per fornire una posizione centrale che riunisce l'amministrazione degli utenti e il loro accesso alle applicazioni Account AWS cloud. Sebbene sia consigliata come best practice per la maggior parte degli utenti con cui lavora AWS, IAM Identity Center attualmente non fornisce meccanismi per credenziali Git o coppie di chiavi SSH. Questi utenti possono installare e configurare gli CodeCommit archivi per git-remote-codecommit clonare localmente, ma non tutti gli ambienti di sviluppo integrati (IDEs) supportano la clonazione, il push o il pulling with. git-remote-codecommit

Come best practice, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere utilizzando credenziali temporanee. Servizi AWS

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

- Ruolo IAM: come un utente IAM, un [ruolo IAM](#) è un'identità IAM che puoi creare nel tuo account per concedere autorizzazioni specifiche.

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Note

Non puoi usare credenziali Git o coppie di chiavi SSH pubbliche-private con gli utenti federati. Inoltre, le preferenze utente non sono disponibili per gli utenti federati. Per informazioni su come configurare le connessioni utilizzando l'accesso federato, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Controllo accessi

È possibile disporre di credenziali valide per autenticare le richieste, ma a meno che non si disponga delle autorizzazioni necessarie non è possibile creare o accedere alle risorse. CodeCommit Ad esempio, devi avere le autorizzazioni necessarie per visualizzare repository, eseguire il push del codice, creare e gestire le credenziali Git e così via.

Le seguenti sezioni descrivono come gestire le autorizzazioni per. CodeCommit Consigliamo di leggere prima la panoramica.

- [Panoramica della gestione delle autorizzazioni di accesso alle risorse CodeCommit](#)
- [Utilizzo di policy basate sull'identità \(IAM Policies\) per CodeCommit](#)
- [Riferimento per le autorizzazioni CodeCommit](#)

Panoramica della gestione delle autorizzazioni di accesso alle risorse CodeCommit

Ogni AWS risorsa è di proprietà di un account Amazon Web Services. Le autorizzazioni per creare o accedere a una risorsa sono regolate dalle policy di autorizzazione. Un amministratore dell'account è in grado di collegare le policy relative alle autorizzazioni alle identità IAM (ovvero utenti, gruppi e ruoli). Alcuni servizi, ad esempio AWS Lambda, supportano anche l'associazione di politiche di autorizzazione alle risorse.

Note

Un amministratore account (o un utente amministratore) è un utente con privilegi di amministratore. Per ulteriori informazioni, consulta [Best practice IAM](#) nella Guida per l'utente di IAM.

Quando si concedono le autorizzazioni, si indicano i destinatari, le risorse a cui si concedono e le operazioni specifiche da consentire su tali risorse.

Argomenti

- [CodeCommit risorse e operazioni](#)
- [Informazioni sulla proprietà delle risorse](#)
- [Gestione dell'accesso alle risorse](#)
- [Ambito delle risorse in CodeCommit](#)
- [Definizione degli elementi delle policy: risorse, operazioni, effetti ed entità principali](#)
- [Specifica delle condizioni in una policy](#)

CodeCommit risorse e operazioni

Nel CodeCommit, la risorsa principale è un repository. A ogni risorsa è associato un Amazon Resource Name (ARN) univoco. In una policy, devi utilizzare un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy stessa. Per ulteriori informazioni su ARNs, consulta

[Amazon Resource Names \(ARN\) e AWS Service Namespaces](#) in. Riferimenti generali di Amazon Web Services CodeCommit attualmente non supporta altri tipi di risorse, denominate sottorisorse.

La tabella seguente descrive come specificare le CodeCommit risorse.

Tipo di risorsa	Formato ARN
Repository	arn:aws:codecommit::: <i>region account-id repository-name</i>
Tutti i CodeCommit repository	arn:aws:codecommit:*
Tutti gli CodeCommit archivi di proprietà dell'account specificato nel campo specificato Regione AWS	arn:aws:codecommit::: * <i>region account-id</i>

Note

La maggior parte AWS dei servizi considera i due punti (:) o una barra (/) come lo stesso carattere. ARNs Tuttavia, CodeCommit richiede una corrispondenza esatta nei modelli e nelle regole delle risorse. Durante la creazione di modelli di eventi, assicurati di utilizzare i caratteri ARN corretti, facendo in modo che corrispondano alla sintassi ARN nella risorsa.

Ad esempio, nella tua dichiarazione puoi specificare un determinato repository (*MyDemoRepo*) utilizzando il relativo ARN come segue:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo"
```

Per specificare tutti i repository che appartengono a un account specifico, usa il carattere jolly (*) come segue:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:*"
```

Per specificare tutte le risorse, o se un'azione API specifica non supporta ARNs, utilizzate il carattere jolly (*) nell'`Resource` elemento come segue:


```
"Resource": "*"
```

Puoi usare il carattere jolly (*) anche per specificare tutte le risorse che corrispondono a una parte del nome di un repository. Ad esempio, il seguente ARN specifica qualsiasi CodeCommit repository che inizia con il nome MyDemo e che è registrato nell'account Amazon Web Services in: 111111111111 us-east-2 Regione AWS

```
arn:aws:codecommit:us-east-2:111111111111:MyDemo*
```

Per un elenco delle operazioni disponibili che funzionano con le CodeCommit risorse, consulta.

[Riferimento per le autorizzazioni CodeCommit](#)

Informazioni sulla proprietà delle risorse

L'account Amazon Web Services possiede le risorse create nell'account, indipendentemente da chi le ha create. In particolare, il proprietario della risorsa è l'account Amazon Web Services dell'[entità principale](#) (ovvero l'account root, un utente IAM o un ruolo IAM) che autentica la richiesta di creazione della risorsa. Negli esempi seguenti viene illustrato il funzionamento:

- Se crei un utente IAM nel tuo account Amazon Web Services e concedi le autorizzazioni per creare CodeCommit risorse a quell'utente, l'utente può creare CodeCommit risorse. Tuttavia, il tuo account Amazon Web Services, a cui appartiene l'utente, possiede le CodeCommit risorse.
- Se utilizzi le credenziali dell'account root del tuo account Amazon Web Services per creare una regola, il tuo account Amazon Web Services è il proprietario della CodeCommit risorsa.
- Se crei un ruolo IAM nel tuo account Amazon Web Services con le autorizzazioni per creare CodeCommit risorse, chiunque possa assumere il ruolo può creare CodeCommit risorse. Il tuo account Amazon Web Services, a cui appartiene il ruolo, possiede le CodeCommit risorse.

Gestione dell'accesso alle risorse

Per gestire l'accesso alle AWS risorse, si utilizzano le politiche di autorizzazione. La policy delle autorizzazioni descrive chi ha accesso a cosa. La sezione che segue spiega le opzioni per la creazione di policy relative alle autorizzazioni.

Note

Questa sezione illustra l'utilizzo di IAM nel contesto di CodeCommit. Non vengono fornite informazioni dettagliate sul servizio IAM. Per ulteriori informazioni su IAM, consulta [Che cos'è](#)

[IAM](#)? nella Guida per l'utente di IAM. Per informazioni sulla sintassi delle policy IAM e le rispettive descrizioni, consulta [Riferimento alle policy IAM di](#) nella Guida per l'utente di IAM.

Le politiche di autorizzazione allegate a un'identità IAM sono denominate politiche basate sull'identità (politiche IAM). Le policy di autorizzazione collegate a una risorsa sono definite policy basate su risorse. Attualmente, CodeCommit supporta solo politiche basate sull'identità (politiche IAM).

Argomenti

- [Policy basate su identità \(policy IAM\)](#)
- [Policy basate sulle risorse](#)

Policy basate su identità (policy IAM)

Per gestire l'accesso alle AWS risorse, alleggi le politiche di autorizzazione alle identità IAM. In CodeCommit, utilizzi policy basate sull'identità per controllare l'accesso ai repository. Ad esempio, puoi eseguire le operazioni seguenti:

- Allega una politica di autorizzazioni a un utente o a un gruppo del tuo account: per concedere a un utente l'autorizzazione a visualizzare le CodeCommit risorse nella CodeCommit console, allega una politica di autorizzazioni basata sull'identità a un utente o gruppo a cui appartiene l'utente.
- Associare un criterio di autorizzazione a un ruolo (per concedere autorizzazioni su più account): la delega, ad esempio quando si desidera concedere l'accesso a più account, implica la creazione di un trust tra l'account proprietario della risorsa (l'account di fiducia) e l'account che contiene gli utenti che devono accedere alla risorsa (l'account affidabile). Una policy di autorizzazione concede all'utente con un ruolo le autorizzazioni necessarie per eseguire le attività previste sulla risorsa. Una policy di attendibilità specifica quali account attendibili possono concedere ai propri utenti le autorizzazioni per assumere il ruolo. [Per ulteriori informazioni, consulta IAM Terms and Concepts.](#)

Per concedere autorizzazioni su più account, collega una policy di autorizzazioni basata sull'identità a un ruolo IAM. Ad esempio, l'amministratore dell'Account A può creare un ruolo per concedere autorizzazioni su più account a un altro account Amazon Web Services (ad esempio, Account B) o a un AWS servizio come segue:

1. L'amministratore dell'account A crea un ruolo IAM e attribuisce una policy di autorizzazione al ruolo che concede le autorizzazioni sulle risorse per l'account A.
2. L'amministratore dell'account A attribuisce una policy di attendibilità al ruolo, identificando l'account B come il principale per tale ruolo.

3. L'amministratore dell'Account B può quindi delegare le autorizzazioni per assumere il ruolo a qualsiasi utente dell'Account B. In questo modo gli utenti dell'Account B possono creare o accedere alle risorse dell'Account A. Se si desidera concedere un'autorizzazione di AWS servizio per assumere il ruolo, il responsabile nella politica di fiducia può anche essere un responsabile del AWS servizio. Per ulteriori informazioni, consulta Delegation in [IAM Terms and Concepts](#).

Per ulteriori informazioni sull'uso di IAM per delegare le autorizzazioni, consulta [Access Management](#) nella IAM User Guide (Guida per l'utente di IAM).

La policy di esempio riportata di seguito consente a un utente di creare un ramo in un repository denominato *MyDemoRepo*:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:CreateBranch"
      ],
      "Resource": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
    }
  ]
}
```

Per limitare le chiamate e le risorse a cui hanno accesso gli utenti del tuo account, crea policy IAM specifiche e poi collega tali policy agli utenti IAM. Per ulteriori informazioni su come creare ruoli IAM e per scoprire esempi di dichiarazioni di policy IAM CodeCommit, consulta [Esempi di politiche di identità gestite dai clienti](#).

Policy basate sulle risorse

Alcuni servizi, come Amazon S3, supportano anche politiche di autorizzazione basate sulle risorse. Ad esempio, puoi allegare una policy basata sulle risorse a un bucket S3 per gestire le autorizzazioni di accesso a quel bucket. CodeCommit non supporta le politiche basate sulle risorse, ma puoi

utilizzare i tag per identificare le risorse, che puoi quindi utilizzare nelle politiche IAM. Per un esempio di policy basate su tag, consulta [Policy basate su identità \(policy IAM\)](#).

Ambito delle risorse in CodeCommit

In CodeCommit, è possibile definire l'ambito delle politiche e delle autorizzazioni basate sull'identità per le risorse, come descritto in [CodeCommit risorse e operazioni](#). Non è però possibile limitare l'ambito dell'autorizzazione `ListRepositories` a una risorsa. Il suo ambito deve invece includere tutte le risorse (usando il carattere jolly `*`). In caso contrario, l'operazione ha esito negativo.

Tutte le altre CodeCommit autorizzazioni possono essere limitate alle risorse.

Definizione degli elementi delle policy: risorse, operazioni, effetti ed entità principali

Puoi creare policy per consentire o negare agli utenti l'accesso alle risorse o consentire o negare agli utenti di intraprendere azioni specifiche su tali risorse. CodeCommit definisce una serie di operazioni API pubbliche che definiscono il modo in cui gli utenti utilizzano il servizio, tramite la CodeCommit console SDKs, il AWS CLI, o chiamandole direttamente. API Per concedere le autorizzazioni per queste operazioni API, CodeCommit definisce una serie di azioni che è possibile specificare in una politica.

Alcune operazioni API possono richiedere le autorizzazioni per più operazioni. Per ulteriori informazioni sulle risorse e sulle operazioni delle API, consulta [CodeCommit risorse e operazioni](#) e [Riferimento per le autorizzazioni CodeCommit](#).

Di seguito sono elencati gli elementi di base di una policy:

- **Risorsa:** per identificare la risorsa a cui si applica la policy, utilizzi un Amazon Resource Name (ARN). Per ulteriori informazioni, consulta [CodeCommit risorse e operazioni](#).
- **Azione:** per identificare le operazioni sulle risorse che desideri consentire o negare, utilizzi parole chiave di azione. Ad esempio, a seconda di quanto specificato `Effect`, l'`codecommit:GetBranch` autorizzazione consente o nega all'utente di eseguire l'`GetBranch` operazione, che consente di ottenere dettagli su una filiale in un CodeCommit repository.
- **Effetto:** si specifica l'effetto, consentito o negato, che si verifica quando l'utente richiede l'azione specifica. `Use` non concedi esplicitamente (consenti) l'accesso a una risorsa, l'accesso viene implicitamente rifiutato. Puoi anche rifiutare esplicitamente l'accesso a una risorsa per fare in modo che un utente non possa accedervi, anche se un'altra policy concede l'accesso.

- **Principal:** nelle politiche basate sull'identità (politiche IAM), l'unico tipo di policy che CodeCommit supporta, l'utente a cui è associata la policy è il principale implicito.

Per ulteriori informazioni sulla sintassi delle policy IAM, consulta IAM [Policy Reference nella IAM User Guide](#).

Per una tabella che mostra tutte le azioni CodeCommit API e le risorse a cui si applicano, consulta [Riferimento per le autorizzazioni CodeCommit](#).

Specifiche delle condizioni in una policy

Quando concedi le autorizzazioni, utilizzi il linguaggio delle policy di accesso per IAM per specificare le condizioni in base alle quali una policy deve avere effetto. Ad esempio, potresti decidere che una policy venga applicata solo dopo una data specifica. Per ulteriori informazioni sulla specificazione delle condizioni in un linguaggio di policy, consulta [Condition](#) and [Policy Grammar](#) nella IAM User Guide.

Per esprimere le condizioni è necessario utilizzare chiavi di condizione predefinite. Non esistono chiavi di condizione specifiche per CodeCommit. Tuttavia, ci sono chiavi di condizione AWS-wide che puoi usare a seconda delle esigenze. Per un elenco completo delle chiavi AWS-wide, consulta [Available Keys for Conditions](#) nella IAM User Guide.

Utilizzo di policy basate sull'identità (IAM Policies) per CodeCommit

I seguenti esempi di policy basate sull'identità dimostrano come un amministratore di account può collegare policy di autorizzazione alle identità IAM (utenti, gruppi e ruoli) per concedere le autorizzazioni per eseguire operazioni sulle risorse. CodeCommit

Important

Ti consigliamo di esaminare innanzitutto gli argomenti introduttivi che spiegano i concetti e le opzioni di base disponibili per gestire l'accesso alle tue risorse. CodeCommit Per ulteriori informazioni, consulta [Panoramica della gestione delle autorizzazioni di accesso alle risorse CodeCommit](#).

Argomenti

- [Autorizzazioni necessarie per l'uso della console CodeCommit](#)

- [Visualizzazione di risorse nella console](#)
- [AWS politiche gestite per CodeCommit](#)
- [Esempi di policy gestite dal cliente](#)

Di seguito è riportato un esempio di policy di autorizzazione basata su identità:

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*"
      ]
    }
  ]
}
```

Questa politica contiene un'unica dichiarazione che consente a un utente di ottenere informazioni sul CodeCommit repository denominato MyDestinationRepo e su tutti i CodeCommit repository che iniziano con il nome MyDemo nella Regione. **us-east-2**

Autorizzazioni necessarie per l'uso della console CodeCommit

Per visualizzare le autorizzazioni richieste per ogni operazione CodeCommit API e per ulteriori informazioni sulle CodeCommit operazioni, vedere. [Riferimento per le autorizzazioni CodeCommit](#)

Per consentire agli utenti di utilizzare la CodeCommit console, l'amministratore deve concedere loro le autorizzazioni per CodeCommit le azioni. Ad esempio, è possibile allegare la politica [AWSCodeCommitPowerUser](#) gestita o il suo equivalente a un utente o gruppo.

Oltre alle autorizzazioni concesse agli utenti dalle politiche basate sull'identità, CodeCommit richiede le autorizzazioni per le azioni (). AWS Key Management Service AWS KMS Un utente IAM non

necessita di Allow autorizzazioni esplicite per queste azioni, ma non deve avere politiche allegate che impostino le seguenti autorizzazioni su: Deny

```
"kms:Encrypt",  
"kms:Decrypt",  
"kms:ReEncrypt",  
"kms:GenerateDataKey",  
"kms:GenerateDataKeyWithoutPlaintext",  
"kms:DescribeKey"
```

Per ulteriori informazioni sulla crittografia e CodeCommit, consulta [AWS KMS e crittografia](#)

Visualizzazione di risorse nella console

La CodeCommit console richiede l'`ListRepositories` autorizzazione per visualizzare un elenco di repository per il tuo account Amazon Web Services nel luogo in Regione AWS cui hai effettuato l'accesso. La console, inoltre, include a funzione Go to resource (Vai alla risorsa) per eseguire rapidamente una ricerca di risorse senza distinzione tra maiuscole e minuscole. Questa ricerca viene eseguita nel tuo account Amazon Web Services nel Regione AWS quale hai effettuato l'accesso. Le seguenti risorse sono visualizzate per i seguenti servizi:

- AWS CodeBuild: progetti di compilazione
- AWS CodeCommit: repository
- AWS CodeDeploy: applicazioni
- AWS CodePipeline: pipeline

Per eseguire la ricerca nelle risorse di tutti i servizi, è necessario disporre delle autorizzazioni seguenti:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

I risultati non vengono restituiti per le risorse di un servizio se non hai le autorizzazioni per quel servizio. Anche se si dispone delle autorizzazioni per la visualizzazione delle risorse, le risorse

specifiche non verranno restituite se è presente un esplicito Deny (Rifiuto) per la visualizzazione di tali risorse.

AWS politiche gestite per CodeCommit

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le policy AWS gestite che scriverle autonomamente. La [creazione di policy gestite dai clienti IAM](#) che forniscono al team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste policy coprono i casi d'uso comuni e sono disponibili nell'account Account AWS. Per ulteriori informazioni sulle policy AWS gestite, consulta le [policy AWS gestite](#) nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consultare la sezione [Policy gestite da AWS per funzioni di processi](#) nella Guida per l'utente di IAM.

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da AWS. Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni. Le policy gestite forniscono CodeCommit anche le autorizzazioni per eseguire operazioni in altri servizi, come IAM, Amazon SNS e CloudWatch Amazon Events, come richiesto per le responsabilità degli utenti a cui è stata concessa la politica in questione. Ad esempio, la AWSCodeCommitFullAccess policy è una policy utente di livello amministrativo che consente agli utenti con questa policy di creare e gestire le regole CloudWatch Events per gli archivi (regole i cui nomi hanno il prefisso da) e gli argomenti di codecommit Amazon SNS per le notifiche sugli eventi relativi ai repository (argomenti i cui nomi hanno il prefisso da), nonché di amministrare i repository in codecommit CodeCommit.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche per CodeCommit.

Argomenti

- [AWS politica gestita: AWSCode CommitFullAccess](#)
- [AWS politica gestita: AWSCode CommitPowerUser](#)
- [AWS politica gestita: AWSCode CommitReadOnly](#)
- [CodeCommit politiche e notifiche gestite](#)
- [AWS CodeCommit politiche gestite e Amazon CodeGuru Reviewer](#)
- [CodeCommit aggiornamenti alle politiche gestite AWS](#)

AWS politica gestita: AWSCode CommitFullAccess

È possibile allegare la policy `AWSCodeCommitFullAccess` alle identità IAM. Questa politica garantisce l'accesso completo a CodeCommit. Applica questa politica solo agli utenti di livello amministrativo a cui desideri concedere il pieno controllo dei CodeCommit repository e delle risorse correlate nel tuo account Amazon Web Services, inclusa la possibilità di eliminare gli archivi.

La `AWSCodeCommitFullAccess` policy contiene la seguente dichiarazione politica:

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 politica gestita: AWSCodeCommitPowerUser

È possibile allegare la policy `AWSCodeCommitPowerUser` alle identità IAM. Questa politica consente agli utenti di accedere a tutte le funzionalità CodeCommit e alle risorse relative al repository, tranne che non consente loro di eliminare gli CodeCommit archivi o creare o eliminare risorse relative ai repository in altri servizi AWS, come Amazon Events. CloudWatch È consigliabile applicare questa policy alla maggior parte degli utenti.

La `AWSCodeCommitPowerUser` policy contiene la seguente dichiarazione politica:

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 politica gestita: AWSCodeCommitReadOnly

È possibile allegare la policy `AWSCodeCommitReadOnly` alle identità IAM. Questa politica garantisce l'accesso in sola lettura CodeCommit e alle risorse relative all'archivio in altri AWS servizi, nonché la possibilità di creare e gestire le proprie risorse CodeCommit correlate (come le credenziali Git e le chiavi SSH che gli utenti IAM possono utilizzare per accedere ai repository). Applica questa policy agli utenti a cui si vuoi consentire di leggere il contenuto di un repository, ma non di modificarlo.

La `AWSCodeCommitReadOnly` policy contiene la seguente dichiarazione politica:

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 politiche e notifiche gestite

AWS CodeCommit supporta le notifiche, che possono notificare agli utenti modifiche importanti agli archivi. Le politiche gestite CodeCommit includono dichiarazioni politiche per la funzionalità di notifica. Per ulteriori informazioni, vedere [Cosa sono le notifiche?](#).

Autorizzazioni correlate alle notifiche nelle policy gestite di accesso completo

Le policy gestite `AWSCodeCommitFullAccess` includono le seguenti dichiarazioni per consentire l'accesso completo alle notifiche. Gli utenti a cui viene applicata questa policy gestita possono anche creare e gestire argomenti Amazon SNS per le notifiche, iscrivere e annullare l'iscrizione degli utenti agli argomenti, elencare argomenti da scegliere come obiettivi per le regole di notifica ed elencare Amazon Q Developer nei client di applicazioni di chat configurati per 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": "*"
}

```

Autorizzazioni correlate alle notifiche nelle policy gestite di sola lettura

Le policy gestite `AWSCodeCommitReadOnlyAccess` includono le seguenti dichiarazioni per consentire l'accesso in sola lettura alle notifiche. Gli utenti con questa policy gestita applicata possono visualizzare le notifiche per le risorse, ma non possono crearle, gestirle o sottoscriverle.

```

{
    "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": "*"
  }

```

Autorizzazioni correlate alle notifiche in altre policy gestite

La policy gestita `AWSCodeCommitPowerUser` include le seguenti istruzioni per consentire agli utenti di creare, modificare e sottoscrivere le notifiche. Gli utenti non possono eliminare le regole di notifica o gestire i tag per le risorse.

```

{
  "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": "*"
  }
}

```

Per ulteriori informazioni su IAM e le notifiche, consulta [Identity and Access Management for AWS CodeStar Notifications](#).

AWS CodeCommit politiche gestite e Amazon CodeGuru Reviewer

CodeCommit supporta Amazon CodeGuru Reviewer, un servizio automatizzato di revisione del codice che utilizza l'analisi dei programmi e l'apprendimento automatico per rilevare problemi comuni e consigliare correzioni nel codice Java o Python. Le politiche gestite CodeCommit includono dichiarazioni politiche per la funzionalità Reviewer. CodeGuru Per ulteriori informazioni, consulta [What Is Amazon CodeGuru Reviewer](#).

Autorizzazioni relative a Reviewer in CodeGuru AWSCode CommitFullAccess

La policy `AWSCodeCommitFullAccess` gestita include le seguenti istruzioni per consentire l'associazione e la dissociazione di CodeGuru Reviewer dai repository. CodeCommit Gli utenti a cui viene applicata questa politica gestita possono anche visualizzare lo stato dell'associazione tra i CodeCommit repository e CodeGuru Reviewer e visualizzare lo stato dei lavori di revisione per le pull request.

```

{
  "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"
      }
    }
  }
}

```

Autorizzazioni relative a Reviewer in CodeGuru AWSCode CommitPowerUser

La policy `AWSCodeCommitPowerUser` gestita include le seguenti istruzioni per consentire agli utenti di associare e dissociare i repository con CodeGuru Reviewer, visualizzare lo stato delle associazioni e visualizzare lo stato dei lavori di revisione per le pull request.

```

{
  "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"
        }
    }
}
}

```

Autorizzazioni relative a Reviewer in CodeGuru AWSCode CommitReadOnly

La policy `AWSCodeCommitReadOnlyAccess` gestita include le seguenti istruzioni per consentire l'accesso in sola lettura allo stato dell'associazione CodeGuru Reviewer e visualizzare lo stato dei lavori di revisione per le pull request. Gli utenti con questa policy gestita applicata non possono associare o disassociare i repository.

```
{
  "Sid": "AmazonCodeGuruReviewerReadOnlyAccess",
  "Effect": "Allow",
  "Action": [
    "codeguru-reviewer:DescribeRepositoryAssociation",
    "codeguru-reviewer:ListRepositoryAssociations",
    "codeguru-reviewer:DescribeCodeReview",
    "codeguru-reviewer:ListCodeReviews"
  ],
  "Resource": "*"
}
```

Ruolo collegato al servizio Amazon CodeGuru Reviewer

Quando si associa un repository a CodeGuru Reviewer, viene creato un ruolo collegato al servizio in modo che CodeGuru Reviewer possa rilevare problemi e consigliare correzioni per il codice Java o Python nelle richieste pull. Il ruolo collegato al servizio viene denominato `AWSServiceRoleForAmazonCodeGuruReviewer`. Per ulteriori informazioni, consulta [Using Service-Linked Roles for Amazon CodeGuru Reviewer](#).

Per ulteriori informazioni, consulta [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

CodeCommit aggiornamenti alle politiche gestite AWS

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CodeCommit da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS su [AWS CodeCommit Cronologia dei documenti della Guida dell'utente](#)

Modifica	Descrizione	Data
AWS politica gestita: AWSCode CommitFullAccess AWS politica gestita: AWSCode CommitPowerUser — Aggiornamento alle politiche esistenti	CodeCommit ha aggiunto un'autorizzazione a queste politiche per supportare un tipo di notifica aggiuntivo o utilizzando Amazon Q Developer nelle applicazioni di chat.	16 maggio 2023

Modifica	Descrizione	Data
	Le AWSCode CommitFullAccess politiche AWSCode CommitPowerUser e sono state modificate per aggiungere un'autorizzazione, chatbot:ListMicrosoftTeamsChannelConfigurations .	
AWS politica gestita: AWSCode CommitReadOnly : aggiornamento di una policy esistente	CodeCommit ha rimosso un'autorizzazione duplicata dalla politica. AWSCodeCommitReadOnly È stata modificata per rimuovere un'autorizzazione duplicata,. "iam:ListAccessKeys"	18 agosto 2021
CodeCommit ha iniziato a tenere traccia delle modifiche	CodeCommit ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	18 agosto 2021

Esempi di policy gestite dal cliente

Puoi creare policy IAM personalizzate per consentire CodeCommit autorizzazioni per azioni e risorse. Puoi associare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono tali autorizzazioni. Puoi anche creare policy IAM personalizzate per l'integrazione tra CodeCommit e altri AWS servizi.

Argomenti

- [Esempi di politiche di identità gestite dai clienti](#)

Esempi di politiche di identità gestite dai clienti

I seguenti esempi di policy IAM concedono autorizzazioni per varie CodeCommit azioni. Utilizzale per limitare CodeCommit l'accesso degli utenti e dei ruoli IAM. Queste policy controllano la capacità di eseguire azioni con la CodeCommit console, l'API o il AWS CLI. AWS SDKs

Note

Tutti gli esempi utilizzano la regione degli Stati Uniti occidentali (Oregon) (us-west-2) e contengono account fittizi. IDs

Esempi

- [Esempio 1: consentire a un utente di eseguire operazioni in un'unica soluzione CodeCommit Regione AWS](#)
- [Esempio 2: consentire a un utente di utilizzare Git per un singolo repository](#)
- [Esempio 3: consentire a un utente che si connette da un intervallo di indirizzi IP specificato di accedere a un repository](#)
- [Esempio 4: negare o consentire azioni sulle filiali](#)
- [Esempio 5: negare o consentire azioni sui repository con tag](#)

Esempio 1: consentire a un utente di eseguire operazioni in un'unica soluzione CodeCommit Regione AWS

La seguente politica di autorizzazione utilizza un carattere jolly ("codecommit:*") per consentire agli utenti di eseguire tutte le CodeCommit azioni nella regione us-east-2 e non da altre. Regioni 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"
            }
        }
    }
]
}

```

Esempio 2: consentire a un utente di utilizzare Git per un singolo repository

In CodeCommit, le autorizzazioni della policy `GitPull` IAM si applicano a qualsiasi comando client Git da cui vengono recuperati i dati CodeCommit, inclusi `git fetch`, `git clone`, e così via. Allo stesso modo, le autorizzazioni della policy `GitPush` IAM si applicano a qualsiasi comando client Git a CodeCommit cui vengono inviati i dati. Ad esempio, se l'autorizzazione della policy `GitPush` IAM è impostata su `Allow`, un utente può inviare l'eliminazione di un ramo utilizzando il protocollo Git. Tale push non è influenzato da alcuna autorizzazione applicata all'`DeleteBranch` operazione per quell'utente IAM. L'`DeleteBranch` autorizzazione si applica alle azioni eseguite con la console, il AWS CLI SDKs, e l'API, ma non il protocollo Git.

L'esempio seguente consente all'utente specificato di estrarre e inviare messaggi dal CodeCommit repository denominato `MyDemoRepo`:

JSON

```

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

```

```
"Effect" : "Allow",
"Action" : [
    "codecommit:GitPull",
    "codecommit:GitPush"
],
"Resource" : "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
}
]
}
```

Esempio 3: consentire a un utente che si connette da un intervallo di indirizzi IP specificato di accedere a un repository

Puoi creare una policy che consente agli utenti di connettersi a un repository CodeCommit se il loro indirizzo IP rientra in un determinato intervallo di indirizzi IP. Sono disponibili due approcci altrettanto validi a questo scopo. È possibile creare una Deny politica che non consenta CodeCommit le operazioni se l'indirizzo IP dell'utente non si trova all'interno di un blocco specifico oppure creare una Allow politica che consenta CodeCommit le operazioni se l'indirizzo IP dell'utente si trova all'interno di un blocco specifico.

Puoi creare una policy Deny che rifiuta l'accesso a tutti gli utenti non inclusi in un determinato intervallo di indirizzi IP. Ad esempio, puoi collegare la policy gestita AWSCodeCommitPowerUser e una policy gestita dal cliente a tutti gli utenti che richiedono l'accesso al repository. La seguente politica di esempio nega tutte le CodeCommit autorizzazioni agli utenti i cui indirizzi IP non rientrano nel blocco di indirizzi IP specificato di 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"
      ]
    }
  }
]
}

```

La seguente politica di esempio consente all'utente specificato di accedere a un CodeCommit archivio denominato MyDemoRepo con le autorizzazioni equivalenti della politica AWSCodeCommitPowerUser gestita solo se il relativo indirizzo IP si trova all'interno del blocco di indirizzi specificato di 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"
          ]
        }
      }
    }
  ]
}

```

```

    ]
  }
}
]
}

```

Esempio 4: negare o consentire azioni sulle filiali

Puoi creare una policy che rifiuta agli utenti le autorizzazioni per eseguire le operazioni specificate su uno o più rami. In alternativa, puoi creare una policy che consente di eseguire operazioni su uno o più rami che altrimenti non sarebbero consentite in altri rami di un repository. Puoi utilizzare queste policy con le policy gestite (predefinite) appropriate. Per ulteriori informazioni, consulta [Limita i push e le unioni ai rami in AWS CodeCommit](#).

Ad esempio, è possibile creare una Deny politica che neghi agli utenti la possibilità di apportare modifiche a un ramo denominato main, inclusa l'eliminazione del ramo, in un repository denominato *MyDemoRepo*. Puoi usare questa policy con la policy gestita `AWSCodeCommitPowerUser`. Gli utenti con queste due politiche applicate sarebbero in grado di creare ed eliminare rami, creare richieste pull e tutte le altre azioni consentite da `AWSCodeCommitPowerUser`, ma non sarebbero in grado di inviare modifiche al ramo denominato main, aggiungere o modificare un file nel ramo principale della CodeCommit console o unire rami o una richiesta pull nel ramo principale. Poiché la policy Deny viene applicata a `GitPush`, devi includervi un'istruzione `Null` per consentire l'analisi della validità delle chiamate `GitPush` iniziali quando gli utenti eseguono push dai loro repository locali.

Tip

Se desideri creare una politica che si applichi a tutti i rami denominati main in tutti i repository del tuo account Amazon Web Services, per `Resource`, specifica un asterisco (*) anziché un ARN del repository.

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"
        }
      }
    }
  ]
}

```

La seguente politica di esempio consente a un utente di apportare modifiche a un ramo denominato main in tutti i repository di un account Amazon Web Services. Non consente modifiche a nessun altro ramo. È possibile utilizzare questa politica con la politica `AWSCodeCommitReadOnly` gestita per consentire push automatici al repository nel ramo principale. Poiché l'effetto è `Allow`, questa policy di esempio non funzionerebbe con le policy gestite come `AWSCodeCommitPowerUser`.

JSON

```

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

```

```

        "Resource": "*",
        "Condition": {
            "StringEqualsIfExists": {
                "codecommit:References": [
                    "refs/heads/main"
                ]
            }
        }
    }
]
}

```

Esempio 5: negare o consentire azioni sui repository con tag

Puoi creare una policy che consenta o neghi azioni sui repository in base ai AWS tag associati a tali repository e quindi applicare tali policy ai gruppi IAM configurati per la gestione degli utenti IAM. Ad esempio, puoi creare una policy che neghi tutte le CodeCommit azioni su qualsiasi repository con il AWS tag key Status e il valore chiave Secret, quindi applicare tale policy al gruppo IAM che hai creato per sviluppatori generici (). *Developers* Devi quindi assicurarti che gli sviluppatori che lavorano su quei repository con tag non siano membri di quel *Developers* gruppo generale, ma appartengano invece a un gruppo IAM diverso a cui non viene applicata la politica restrittiva (). SecretDevelopers

L'esempio seguente nega tutte le CodeCommit azioni sui repository contrassegnati con la chiave Status e il valore chiave di 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"
    }
  }
}
```

È possibile perfezionare ulteriormente questa strategia specificando repository specifici, anziché tutti i repository, come risorse. È inoltre possibile creare politiche che consentano CodeCommit azioni su tutti gli archivi che non sono etichettati con tag specifici. Ad esempio, la seguente politica consente l'equivalente delle AWSCodeCommitPowerUser CodeCommit autorizzazioni per le azioni, tranne per il fatto che consente solo CodeCommit azioni su repository non contrassegnati con i tag specificati:

Note

Questo esempio di politica include solo azioni per. CodeCommit Non include azioni per altri AWS servizi inclusi nella politica AWSCodeCommitPowerUser gestita. Per ulteriori informazioni, consulta [AWS politica gestita: AWSCode CommitPowerUser..](#)

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": "*"
}
```

Riferimento per le autorizzazioni CodeCommit

Le tabelle seguenti elencano ogni operazione CodeCommit API, le azioni corrispondenti per le quali è possibile concedere le autorizzazioni e il formato della risorsa ARN da utilizzare per la concessione delle autorizzazioni. CodeCommit APIs Sono raggruppate in tabelle in base all'ambito delle azioni consentite da tale API. Fai riferimento ad essa quando configuri [Controllo accessi](#) e scrivi le politiche di autorizzazione da allegare a un'identità IAM (politiche basate sull'identità).

Quando si crea una policy di autorizzazione, è necessario specificare le operazioni nel campo Action della policy. Puoi specificare il valore della risorsa nel campo Resource della policy come ARN, con o senza un carattere jolly (*).

Per esprimere le condizioni nelle tue CodeCommit policy, usa AWS le chiavi di condizione -wide. Per un elenco completo delle chiavi AWS-wide, consulta [Available Keys](#) nella IAM User Guide. Per informazioni complete su azioni, risorse e chiavi di condizione per CodeCommit le policy IAM, consulta [Actions, resources and condition keys for AWS CodeCommit](#).

Note

Per specificare un'operazione, utilizza il prefisso `codecommit:` seguito dal nome dell'operazione API (ad esempio, `codecommit:GetRepository` o `codecommit:CreateRepository`).

Utilizzo di caratteri jolly

Per specificare più operazioni o risorse, usa un carattere jolly (*) nell'ARN. Ad esempio, `codecommit:*` specifica tutte le CodeCommit azioni e `codecommit:Get*` specifica tutte le CodeCommit azioni che iniziano con la parola. Get L'esempio seguente concede l'accesso completo a tutti i repository con nomi che iniziano con MyDemo:

```
arn:aws:codecommit:us-west-2:111111111111:MyDemo*
```

È possibile utilizzare i caratteri jolly solo con le *repository-name* risorse elencate nella tabella seguente. Non puoi usare i caratteri jolly con le nostre risorse *region*. *account-id* Per ulteriori informazioni sui caratteri jolly, consulta [IAM Identifiers nella IAM User Guide](#).

Argomenti

- [Autorizzazioni richieste per i comandi client Git](#)
- [Autorizzazioni per le azioni sulle filiali](#)
- [Autorizzazioni per le azioni sulle unioni](#)
- [Autorizzazioni per le azioni sulle richieste pull](#)
- [Autorizzazioni per le azioni sui modelli di regole di approvazione](#)
- [Autorizzazioni per azioni su singoli file](#)
- [Autorizzazioni per le azioni sui commenti](#)
- [Autorizzazioni per azioni sul codice commesso](#)
- [Autorizzazioni per le azioni sui repository](#)
- [Autorizzazioni per le azioni sui tag](#)
- [Autorizzazioni per le azioni sui trigger](#)
- [Autorizzazioni per azioni di integrazione CodePipeline](#)

Autorizzazioni richieste per i comandi client Git

In CodeCommit, le autorizzazioni della policy `GitPull` IAM si applicano a qualsiasi comando client Git da cui vengono recuperati i dati CodeCommit, inclusi `git fetch` `git clone`, e così via. Allo stesso modo, le autorizzazioni della policy `GitPush` IAM si applicano a qualsiasi comando client Git a CodeCommit cui vengono inviati i dati. Ad esempio, se l'autorizzazione della policy `GitPush` IAM è impostata su `Allow`, un utente può inviare l'eliminazione di un ramo utilizzando il protocollo Git.

Tale push non è influenzato da alcuna autorizzazione applicata all>DeleteBranchoperazione per quell'utente IAM. L>DeleteBranchautorizzazione si applica alle azioni eseguite con la console, il AWS CLI SDKs, e l'API, ma non il protocollo Git.

GitPull e GitPush sono autorizzazioni relative alle policy IAM. Non sono operazioni API.

CodeCommit Autorizzazioni richieste per le azioni relative ai comandi Git Client

GitPull

Operazioni: `codecommit:GitPull`

Necessario per estrarre informazioni da un CodeCommit repository a un repository locale. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GitPush

Operazioni: `codecommit:Git Push`

Necessario per inviare informazioni da un repository locale a un repository. CodeCommit Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sulle filiali

Le seguenti autorizzazioni consentono o negano azioni sulle filiali negli archivi. CodeCommit Queste autorizzazioni riguardano solo le azioni eseguite nella CodeCommit console e con l' CodeCommit API e i comandi eseguiti utilizzando. AWS CLI Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Ad esempio, il comando `git show-branch -r` visualizza un elenco dei rami remoti per un repository e i relativi commit utilizzando il protocollo Git. Non è influenzata da alcuna autorizzazione per l'operazione. CodeCommit `ListBranches`

Per ulteriori informazioni sulle politiche per le filiali, consulta [Limita i push e le unioni ai rami in AWS CodeCommit](#) e [Esempi di policy gestite dal cliente](#).

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sulle filiali

[CreateBranch](#)

Operazioni: `codecommit:CreateBranch`

Necessario per creare una filiale in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[DeleteBranch](#)

Operazioni: `codecommit:DeleteBranch`

Necessario per eliminare un ramo da un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetBranch](#)

Operazioni: `codecommit:GetBranch`

Necessario per ottenere dettagli su un ramo in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[ListBranches](#)

Operazioni: `codecommit:ListBranches`

Necessario per ottenere un elenco di filiali in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesByFastForward](#)

Operazioni: `codecommit:MergeBranchesByFastForward`

Necessario per unire due rami utilizzando la strategia di fusione rapida in un repository.
CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesBySquash](#)

Operazioni: `codecommit:ListBranches`

Richiesto per unire due rami utilizzando la strategia di unione squash in un repository
CodeCommit .

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

MergeBranchesByThreeWay

Operazioni: `codecommit:ListBranches`

Obbligatorio per unire due rami utilizzando la strategia di unione a tre vie in un repository CodeCommit .

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UpdateDefaultBranch

Operazioni: `codecommit:UpdateDefaultBranch`

Necessario per modificare il ramo predefinito in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sulle unioni

Le seguenti autorizzazioni consentono o negano azioni sulle unioni nei repository. CodeCommit Queste autorizzazioni riguardano le azioni eseguite con la CodeCommit console e l' CodeCommitAPI e i comandi eseguiti utilizzando. AWS CLI Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Per le autorizzazioni correlate su rami, consulta [Autorizzazioni per le azioni sulle filiali](#). Per le autorizzazioni correlate per le richieste di pull, consulta [Autorizzazioni per le azioni sulle richieste pull](#).

CodeCommit Operazioni API e autorizzazioni richieste per le azioni per i comandi di unione

BatchDescribeMergeConflicts

Operazioni: `codecommit:BatchDescribeMergeConflicts`

Necessario per restituire informazioni sui conflitti in un'unione tra commit in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

CreateUnreferencedMergeCommit

Operazioni: `codecommit:CreateUnreferencedMergeCommit`

Necessario per creare un commit non referenziato tra due branch o commit in un CodeCommit repository allo scopo di confrontarli e identificare eventuali conflitti.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

DescribeMergeConflicts

Operazioni: `codecommit:DescribeMergeConflicts`

Richiesto per restituire informazioni sui conflitti di unione tra le versioni di base, di origine e di destinazione di un file in una unione potenziale in un repository CodeCommit .

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeCommit

Operazioni: `codecommit:GetMergeCommit`

Necessario per restituire informazioni sull'unione tra un commit di origine e uno di destinazione in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeOptions

Operazioni: `codecommit:GetMergeOptions`

Obbligatorio per restituire informazioni sulle opzioni di unione disponibili tra due branch o specificatori di commit in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sulle richieste pull

Le seguenti autorizzazioni permettono o rifiutano le richieste pull nei repository CodeCommit. Queste autorizzazioni riguardano le azioni eseguite con la CodeCommit console e l' CodeCommit API e i comandi eseguiti utilizzando. AWS CLI Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Per le autorizzazioni correlate sui commenti, consulta [Autorizzazioni per le azioni sui commenti](#).

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sulle richieste pull

BatchGetPullRequests

Operazioni: `codecommit:BatchGetPullRequests`

Necessaria per restituire informazioni su una o più richieste pull in un repository CodeCommit . Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequest

Operazioni: `codecommit:CreatePullRequest`

Necessario per creare una pull request in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequestApprovalRule

Operazioni: `codecommit:CreatePullRequestApprovalRule`

Necessario per creare una regola di approvazione per una richiesta pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

DeletePullRequestApprovalRule

Operazioni: `codecommit>DeletePullRequestApprovalRule`

Necessario per eliminare una regola di approvazione per una richiesta pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

DescribePullRequestEvents

Operazioni: `codecommit:DescribePullRequestEvents`

Necessario per restituire informazioni su uno o più eventi di pull request in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

EvaluatePullRequestApprovalRules

Operazioni: `codecommit:EvaluatePullRequestApprovalRules`

Necessario per valutare se una pull request ha soddisfatto tutte le condizioni specificate nelle regole di approvazione associate in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentsForPullRequest

Operazioni: `codecommit:GetCommentsForPullRequest`

Necessaria per restituire i commenti effettuati su una richiesta di pull.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitsFromMergeBase

Operazioni: `codecommit:GetCommitsFromMergeBase`

Necessaria per restituire informazioni sulla differenza tra i commit nel contesto di una potenziale unione. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeConflicts

Operazioni: `codecommit:GetMergeConflicts`

Necessario per restituire informazioni sui conflitti di fusione tra il ramo di origine e quello di destinazione in una pull request.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequest

Operazioni: `codecommit:GetPullRequest`

Necessaria per restituire informazioni su una richiesta pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequestApprovalStates

Operazioni: `codecommit:GetPullRequestApprovalStates`

Necessario per restituire informazioni sugli stati di approvazione per una richiesta pull specificata.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetPullRequestOverrideState

Operazioni: `codecommit:GetPullRequestOverrideState`

Obbligatorio per indicare se le regole di approvazione sono state messe da parte (sostituite) per una pull request e, in caso affermativo, l'Amazon Resource Name (ARN) dell'utente o dell'identità che ha annullato le regole e i relativi requisiti per la pull request.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[ListPullRequests](#)

Operazioni: `codecommit:ListPullRequests`

Necessaria per elencare le richieste pull in un repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[MergePullRequestByFastForward](#)

Operazioni: `codecommit:MergePullRequestByFastForward`

Necessaria per chiudere una richiesta di pull e tentare di unire il ramo di origine al ramo di destinazione di una richiesta di pull utilizzando la strategia di unione avanzamento rapido.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[MergePullRequestBySquash](#)

Operazioni: `codecommit:MergePullRequestBySquash`

Necessaria per chiudere una richiesta di pull e tentare di unire il ramo di origine al ramo di destinazione di una richiesta di pull utilizzando la strategia di unione squash.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[MergePullRequestByThreeWay](#)

Operazioni: `codecommit:MergePullRequestByThreeWay`

Necessaria per chiudere una richiesta di pull e tentare di unire il ramo di origine al ramo di destinazione di una richiesta di pull utilizzando la strategia di unione a tre vie.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[OverridePullRequestApprovalRules](#)

Operazioni: `codecommit:OverridePullRequestApprovalRules`

Necessario per mettere da parte tutti i requisiti delle regole di approvazione per una pull request in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

PostCommentForPullRequest

Operazioni: `codecommit:PostCommentForPullRequest`

Necessaria per pubblicare un commento su una richiesta di pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestApprovalRuleContent

Operazioni: `codecommit:UpdatePullRequestApprovalRuleContent`

Necessario per modificare la struttura di una regola di approvazione per una pull request in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestApprovalState

Operazioni: `codecommit:UpdatePullRequestApprovalState`

Necessario per aggiornare lo stato di un'approvazione su una richiesta pull in un repository CodeCommit .

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestDescription

Operazioni: `codecommit:UpdatePullRequestDescription`

Necessaria per modificare la descrizione di una richiesta di pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestStatus

Operazioni: `codecommit:UpdatePullRequestStatus`

Necessaria per modificare lo stato di una richiesta di pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[UpdatePullRequestTitle](#)

Operazioni: `codecommit:UpdatePullRequestTitle`

Necessaria per modificare il titolo di una richiesta di pull in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sui modelli di regole di approvazione

Le seguenti autorizzazioni consentono o negano operazioni sui modelli di regola di approvazione nei repository CodeCommit . Queste autorizzazioni riguardano solo le azioni eseguite nella CodeCommit console, nell' CodeCommit API e i comandi eseguiti utilizzando AWS CLI. Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Per le autorizzazioni correlate per le richieste di pull, consulta [Autorizzazioni per le azioni sulle richieste pull](#).

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sui modelli di regole di approvazione

[AssociateApprovalRuleTemplateWithRepository](#)

Operazioni: `codecommit:AssociateApprovalRuleTemplateWithRepository`

Necessario per associare un modello a un repository specificato in un account Amazon Web Services. Una volta associato, questo crea automaticamente regole di approvazione che corrispondono alle condizioni del modello per ogni richiesta pull creata nel repository specificato.

Risorsa: *

[BatchAssociateApprovalRuleTemplateWithRepositories](#)

Operazioni: `codecommit:BatchAssociateApprovalRuleTemplateWithRepositories`

Necessario per associare un modello a uno o più repository specificati in un account Amazon Web Services.

Risorsa: *

[BatchDisassociateApprovalRuleTemplateFromRepositories](#)

Operazioni:

`codecommit:BatchDisassociateApprovalRuleTemplateFromRepositories`

Necessario per dissociare un modello da uno o più repository specificati in un account Amazon Web Services.

Risorsa: *

[CreateApprovalRuleTemplate](#)

Operazioni: `codecommit:CreateApprovalRuleTemplate`

Necessario per creare un modello per le regole di approvazione che può quindi essere associato a uno o più repository nel tuo account Amazon Web Services.

Risorsa: *

[DeleteApprovalRuleTemplate](#)

Operazioni: `codecommit>DeleteApprovalRuleTemplate`

Necessario per eliminare un modello di regola di approvazione da un account AWS .

Risorsa: *

[DisassociateApprovalRuleTemplateFromRepository](#)

Operazioni: `codecommit:DisassociateApprovalRuleTemplateFromRepository`

Necessario per dissociare il modello specificato da un repository in un account Amazon Web Services. Non rimuove le regole di approvazione sulle richieste pull già create con il modello.

Risorsa: *

[GetApprovalRuleTemplate](#)

Operazioni: `codecommit:GetApprovalRuleTemplate`

Necessario per restituire informazioni su un modello di regola di approvazione in un account Amazon Web Services.

Risorsa: *

[ListApprovalRuleTemplates](#)

Operazioni: `codecommit:ListApprovalRuleTemplates`

Necessario per elencare i modelli di regole di approvazione in un account Amazon Web Services.

Risorsa: *

[ListAssociatedApprovalRuleTemplatesForRepository](#)

Operazioni: `codecommit:ListAssociatedApprovalRuleTemplatesForRepository`

Necessario per elencare tutti i modelli di regole di approvazione associati a un repository specificato in un account Amazon Web Services.

Risorsa: *

[ListRepositoriesForApprovalRuleTemplate](#)

Operazioni: `codecommit:ListRepositoriesForApprovalRuleTemplate`

Necessario per elencare tutti i repository associati a un modello di regola di approvazione specificato in un account Amazon Web Services.

Risorsa: *

[UpdateApprovalRuleTemplateContent](#)

Operazioni: `codecommit:UpdateApprovalRuleTemplateContent`

Necessario per aggiornare il contenuto di un modello di regola di approvazione in un account Amazon Web Services.

Risorsa: *

[UpdateApprovalRuleTemplateDescription](#)

Operazioni: `codecommit:UpdateApprovalRuleTemplateDescription`

Necessario per aggiornare la descrizione di un modello di regola di approvazione in un account Amazon Web Services.

Risorsa: *

[UpdateApprovalRuleTemplateName](#)

Operazioni: `codecommit:UpdateApprovalRuleTemplateName`

Necessario per aggiornare il nome di un modello di regola di approvazione in un account AWS .

Risorsa: *

Autorizzazioni per azioni su singoli file

Le seguenti autorizzazioni permettono o rifiutano le operazioni su singoli file nei repository CodeCommit. Queste autorizzazioni riguardano solo le azioni eseguite nella CodeCommit console, nell' CodeCommit API e i comandi eseguiti utilizzando AWS CLI. Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Ad esempio, il comando `git push` esegue il push dei file nuovi e modificati in un repository CodeCommit utilizzando il protocollo Git. Non è influenzata da alcuna autorizzazione per l'operazione. CodeCommit PutFile

CodeCommit Operazioni API e autorizzazioni richieste per le azioni su singoli file

[DeleteFile](#)

Operazioni: `codecommit:DeleteFile`

Necessario per eliminare un file specificato da un ramo specifico in un CodeCommit repository dalla CodeCommit console.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetBlob](#)

Operazioni: `codecommit:GetBlob`

Necessario per visualizzare il contenuto codificato di un singolo file in un CodeCommit repository dalla console. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetFile](#)

Operazioni: `codecommit:GetFile`

Necessario per visualizzare il contenuto codificato di un file specificato e i relativi metadati in un CodeCommit repository dalla console. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetFolder](#)

Operazioni: `codecommit:GetFolder`

Necessario per visualizzare il contenuto di una cartella specificata in un CodeCommit repository dalla console. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

PutFile

Operazioni: `codecommit:PutFile`

Necessario per aggiungere un file nuovo o modificato a un CodeCommit repository dalla CodeCommit console, dall' CodeCommit API o dal. AWS CLI

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sui commenti

Le seguenti autorizzazioni consentono o negano azioni sui commenti negli archivi. CodeCommit Queste autorizzazioni riguardano le azioni eseguite con la CodeCommit console e l' CodeCommitAPI e i comandi eseguiti utilizzando. AWS CLI Per le autorizzazioni correlate sui commenti nelle richieste pull, consulta [Autorizzazioni per le azioni sulle richieste pull](#).

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sui repository

DeleteCommentContent

Operazioni: `codecommit>DeleteCommentContent`

Necessaria per eliminare il contenuto di un commento effettuato su una modifica, un file o un commit in un repository. I commenti non possono essere eliminati, ma il contenuto di un commento può essere rimosso se l'utente dispone di questa autorizzazione.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetComment

Operazioni: `codecommit:GetComment`

Necessario per restituire informazioni su un commento fatto su una modifica, un file o un commit in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentReactions

Operazioni: `codecommit:GetCommentReactions`

Obbligatorio per restituire informazioni sulle reazioni emoji a un commento fatto su una modifica, un file o un commit in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommentsForComparedCommit](#)

Operazioni: `codecommit:GetCommentsForComparedCommit`

Obbligatorio per restituire informazioni sui commenti fatti sul confronto tra due commit in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[PostCommentForComparedCommit](#)

Operazioni: `codecommit:PostCommentForComparedCommit`

Necessaria per creare un commento sul confronto tra due commit in un repository CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[PostCommentReply](#)

Operazioni: `codecommit:PostCommentReply`

Necessario per creare una risposta a un commento su un confronto tra commit o su una pull request in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[PutCommentReaction](#)

Operazioni: `codecommit:PutCommentReaction`

Necessario per rispondere a un commento con un'emoji su un commit o su una pull request in un repository. CodeCommit

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[UpdateComment](#)

Operazioni: `codecommit:UpdateComment`

Necessaria per modificare un commento su un confronto tra commit o su una richiesta di pull. I commenti possono essere modificati solo dall'autore.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per azioni sul codice commesso

Le seguenti autorizzazioni permettono o rifiutano le operazioni sul codice sottoposto a commit nei repository CodeCommit. Queste autorizzazioni riguardano le azioni eseguite con la CodeCommit console e l' CodeCommit API e i comandi eseguiti utilizzando AWS CLI. Non riguardano operazioni simili che possono essere eseguite con il protocollo Git. Ad esempio, il comando `git commit` crea un commit per un ramo in un repository utilizzando il protocollo Git. Non è influenzata da alcuna autorizzazione per l'operazione `CreateCommit`.

La negazione esplicita di alcune di queste autorizzazioni potrebbe comportare conseguenze impreviste sulla console. CodeCommit. Ad esempio, se si imposta `GetTree` su `Deny` si impedisce agli utenti di esplorare il contenuto di un repository nella console, ma non di visualizzare il contenuto di un file nel repository (ad esempio se ricevono un messaggio e-mail contenente un link al file). Se si imposta `GetBlob` su `Deny` si impedisce agli utenti di visualizzare il contenuto dei file, ma non di esplorare la struttura di un repository. Se si imposta `GetCommit` su `Deny` si impedisce agli utenti di recuperare dettagli sui commit. Se si imposta `GetObjectIdentifier` su `Deny` si blocca la maggior parte delle funzionalità di esplorazione del codice. Se imposti tutte e tre queste azioni `Deny` in una policy, un utente con quella policy non può sfogliare il codice nella console. CodeCommit.

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sul codice commesso

BatchGetCommits

Operazioni: `codecommit:BatchGetCommits`

Necessaria per restituire informazioni su uno o più commit in un repository CodeCommit. Questa è solo un'autorizzazione della policy IAM, non un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[CreateCommit](#)

Operazioni: `codecommit>CreateCommit`

Obbligatoria per creare un commit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommit](#)

Operazioni: `codecommit:GetCommit`

Necessaria per restituire informazioni su un commit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitHistory

Operazioni: `codecommit:GetCommitHistory`

Necessaria per restituire informazioni sulla cronologia dei commit in un repository. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetDifferences](#)

Operazioni: `codecommit:GetDifferences`

Necessaria per restituire informazioni sulle differenze in uno specificatore di commit (ad esempio un ramo, tag, HEAD, ID commit o altri riferimenti completi).

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetObjectIdentifier

Operazioni: `codecommit:GetObjectIdentifier`

Necessaria per risolvere BLOB, alberi e commit nel relativo identificatore. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetReferences

Operazioni: `codecommit:GetReferences`

Necessaria per restituire tutti i riferimenti, ad esempio rami e tag. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

GetTree

Operazioni: `codecommit:GetTree`

Necessario per visualizzare il contenuto di un albero specificato in un CodeCommit repository dalla CodeCommit console. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sui repository

Le seguenti autorizzazioni consentono o negano azioni sui repository. CodeCommit Queste autorizzazioni riguardano le azioni eseguite con la CodeCommit console e l' CodeCommit API e i comandi eseguiti utilizzando. AWS CLI Non riguardano operazioni simili che possono essere eseguite con il protocollo Git.

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sui repository

[BatchGetRepositories](#)

Operazioni: `codecommit:BatchGetRepositories`

Necessario per ottenere informazioni su più CodeCommit repository presenti in un account Amazon Web Services. InResource, devi specificare i nomi di tutti i CodeCommit repository per i quali a un utente sono consentite (o negate) le informazioni.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[CreateRepository](#)

Operazioni: `codecommit>CreateRepository`

Necessario per creare un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[DeleteRepository](#)

Operazioni: `codecommit>DeleteRepository`

Necessario per eliminare un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetRepository](#)

Operazioni: `codecommit:GetRepository`

Necessario per ottenere informazioni su un singolo CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[ListRepositories](#)

Operazioni: `codecommit:ListRepositories`

Necessario per ottenere un elenco dei nomi e IDs del sistema di più CodeCommit repository per un account Amazon Web Services. Il solo valore consentito per `Resource` per questa operazione è tutti i repository (*).

Risorsa: *

[UpdateRepositoryDescription](#)

Operazioni: `codecommit:UpdateRepositoryDescription`

Necessario per modificare la descrizione di un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[UpdateRepositoryName](#)

Operazioni: `codecommit:UpdateRepositoryName`

Necessario per modificare il nome di un CodeCommit repository. In `Resource`, è necessario specificare sia i CodeCommit repository che possono essere modificati sia i nuovi nomi dei repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sui tag

Le seguenti autorizzazioni consentono o negano azioni sui AWS tag delle risorse. CodeCommit

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sui tag

[ListTagsForResource](#)

Operazioni: `codecommit:ListTagsForResource`

Necessario per restituire informazioni sui AWS tag configurati su una risorsa in CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[TagResource](#)

Operazioni: `codecommit:TagResource`

Necessario per aggiungere o modificare i AWS tag per un repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

UntagResource

Operazioni: `codecommit:UntagResource`

Necessario per rimuovere i AWS tag da una risorsa in CodeCommit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per le azioni sui trigger

Le seguenti autorizzazioni permettono o rifiutano le operazioni sui trigger per i repository CodeCommit.

CodeCommit Operazioni API e autorizzazioni richieste per le azioni sui trigger

GetRepositoryTriggers

Operazioni: `codecommit:GetRepositoryTriggers`

Necessaria per restituire informazioni sui trigger configurati per un repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

PutRepositoryTriggers

Operazioni: `codecommit:PutRepositoryTriggers`

Necessaria per creare, modificare o eliminare i trigger per un repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

TestRepositoryTriggers

Operazioni: `codecommit:TestRepositoryTriggers`

Necessaria per testare le funzionalità di un trigger di repository mediante l'invio di dati all'argomento o alla funzione configurata per il trigger.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Autorizzazioni per azioni di integrazione CodePipeline

CodePipeline Per utilizzare un CodeCommit repository in un'azione di origine per una pipeline, è necessario concedere tutte le autorizzazioni elencate nella tabella seguente al ruolo di servizio per. CodePipeline Se queste autorizzazioni non sono impostate nel ruolo del servizio o vengono impostate su **Deny**, la pipeline non viene eseguita automaticamente quando viene apportata una modifica al repository e non è possibile rilasciare manualmente le modifiche.

CodeCommit Operazioni API e autorizzazioni richieste per le azioni di integrazione CodePipeline

[GetBranch](#)

Operazioni: `codecommit:GetBranch`

Necessario per ottenere dettagli su una filiale in un CodeCommit repository.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetCommit](#)

Operazioni: `codecommit:GetCommit`

Necessaria per restituire informazioni su un commit.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[UploadArchive](#)

Operazioni: `codecommit:UploadArchive`

Necessario per consentire al ruolo di servizio CodePipeline di caricare le modifiche al repository in una pipeline. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

[GetUploadArchiveStatus](#)

Operazioni: `codecommit:GetUploadArchiveStatus`

Necessaria per determinare lo stato di un caricamento di archivio, ovvero se è in corso, completato, annullato o se si è verificato un errore. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

CancelUploadArchive

Operazioni: `codecommit:CancelUploadArchive`

Necessaria per annullare il caricamento di un archivio in una pipeline. Si tratta solo di un'autorizzazione della policy IAM, non di un'azione API che puoi chiamare.

Risorsa: `arn:aws:codecommit:region:account-id:repository-name`

Come AWS CodeCommit funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a CodeCommit, è necessario comprendere con quali funzionalità IAM è disponibile l'uso CodeCommit. Per avere una visione di alto livello di come CodeCommit e altri AWS servizi funzionano con IAM, consulta [AWS Services That Work with IAM nella IAM User Guide](#).

Argomenti

- [Chiavi di condizione](#)
- [Esempi](#)

Chiavi di condizione

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

CodeCommit definisce il proprio set di chiavi di condizione e supporta anche l'utilizzo di alcune chiavi di condizione globali. Per visualizzare tutte le chiavi di condizione AWS globali, consulta [AWS Global Condition Context Keys](#) nella IAM User Guide.

Alcune CodeCommit azioni supportano la chiave di `codecommit:References` condizione. Per una policy di esempio che utilizza questa chiave, consulta [Esempio 4: negare o consentire azioni sulle filiali](#).

Per visualizzare un elenco di chiavi di CodeCommit condizione, consulta [Condition Keys for AWS CodeCommit](#) nella IAM User Guide. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Azioni definite da AWS CodeCommit](#).

Esempi

Per visualizzare esempi di politiche CodeCommit basate sull'identità, vedere. [AWS CodeCommit esempi di politiche basate sull'identità](#)

Policy di CodeCommit basate sulle risorse

CodeCommit non supporta politiche basate sulle risorse.

Autorizzazione basata su tag CodeCommit

È possibile allegare tag alle CodeCommit risorse o passare tag in una richiesta a CodeCommit. Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `codecommit:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Per ulteriori informazioni sull'etichettatura CodeCommit delle risorse, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#). Per ulteriori informazioni sulle strategie di etichettatura, consulta [AWS Tagging Resources](#).

CodeCommit supporta anche politiche basate sui tag di sessione. Per ulteriori informazioni, consulta la sezione relativa ai [tag di sessione](#).

Utilizzo dei tag per fornire informazioni sull'identità in CodeCommit

CodeCommit supporta l'uso di tag di sessione, che sono attributi di coppia chiave-valore che vengono trasmessi quando si assume un ruolo IAM, si utilizzano credenziali temporanee o si federa un utente in (). AWS Security Token Service AWS STS È inoltre possibile associare tag a un utente IAM. Puoi utilizzare le informazioni fornite in questi tag per identificare più facilmente chi ha apportato una modifica o causato un evento. CodeCommit include i valori per i tag con i seguenti nomi chiave negli CodeCommit eventi:

Nome chiave	Valore
<code>displayName</code>	Il nome in formato leggibile da visualizzare e associare all'utente (ad esempio, Mary Major o Saanvi Sarkar).

Nome chiave	Valore
emailAddress	L'indirizzo e-mail che si desidera visualizzare e associare all'utente (ad esempio, mary_major@example.com o saanvi_sarkar@example.com).

Se queste informazioni vengono fornite, le CodeCommit include negli eventi inviati ad Amazon EventBridge e Amazon CloudWatch Events. Per ulteriori informazioni, consulta [Monitoraggio CodeCommit degli eventi in Amazon EventBridge e Amazon CloudWatch Events](#).

Per utilizzare il tagging di sessione, i ruoli devono disporre di policy che includano l'autorizzazione `sts:TagSession` impostata su `Allow`. Se si utilizza l'accesso federato, è possibile configurare le informazioni relative al nome visualizzato e al tag di e-mail come parte della configurazione. Ad esempio, se si utilizza Azure Active Directory, è possibile fornire le seguenti informazioni di registrazione:

Nome di registrazione	Valore
<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>

Puoi usare il AWS CLI per passare i tag di sessione `displayName` e `emailAddress` utilizzarli `AssumeRole`. Ad esempio, un utente che desidera assumere un ruolo denominato *Developer* chi desidera associare il proprio nome *Mary Major* potrebbe utilizzare il `assume-role` comando simile al seguente:

```
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
```

Per ulteriori informazioni, consulta [AssumeRole](#).

È possibile utilizzare l'operazione `AssumeRoleWithSAML` per restituire un set di credenziali temporanee che includano i tag `displayName` e `emailAddress`. È possibile utilizzare questi tag quando si accede ai repository CodeCommit. Ciò richiede che l'azienda o il gruppo abbiano già integrato la soluzione SAML di terze parti con AWS. In tal caso, è possibile passare gli attributi SAML come tag di sessione. Ad esempio, se desideri passare gli attributi di identità per un nome visualizzato e un indirizzo e-mail per un utente *Saanvi Sarkar* denominato tag di sessione:

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:displayName">
  <AttributeValue>Saanvi Sarkar</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:emailAddress">
  <AttributeValue>saanvi_sarkar@example.com</AttributeValue>
</Attribute>
```

Per maggiori informazioni, consulta [Passare i tag di sessione utilizzando AssumeRoleWith SAML](#).

È possibile utilizzare l'operazione `AssumeRoleWithIdentity` per restituire un set di credenziali temporanee che includano i tag `displayName` e `emailAddress`. È possibile utilizzare questi tag quando si accede ai repository CodeCommit. Per passare i tag di sessione da OpenID Connect (OIDC), è necessario includere i tag di sessione nel Token Web JSON (JWT). Ad esempio, il token JWP decodificato utilizzato per la chiamata include `AssumeRoleWithWebIdentity` i tag `displayName` e `emailAddress` session per un utente denominato: *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"
    ]
}
}
```

Per ulteriori informazioni, vedete [Passare i tag di sessione utilizzando](#). AssumeRoleWithWebIdentity

È possibile utilizzare l'operazione GetFederationToken per restituire un set di credenziali temporanee che includano i tag displayName e emailAddress. È possibile utilizzare questi tag quando si accede ai repository CodeCommit. Ad esempio, per utilizzare il AWS CLI per ottenere un token di federazione che includa i emailAddress tag displayName and:

```
aws sts get-federation-token \
--name my-federated-user \
--tags key=displayName,value="Nikhil Jayashankar"
      key=emailAddress,value=nikhil_jayashankar@example.com
```

Per ulteriori informazioni, consulta [Passare i tag di sessione utilizzando GetFederationToken](#).

CodeCommit Ruoli IAM

Un [ruolo IAM](#) è un'entità all'interno dell'account Amazon Web Services che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con CodeCommit

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come [AssumeRole](#)o. [GetFederationToken](#)

CodeCommit supporta l'utilizzo di credenziali temporanee. Per ulteriori informazioni, consulta [Connessione ai AWS CodeCommit repository con credenziali rotanti](#).

Ruoli collegati ai servizi

[I ruoli collegati ai](#) AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

CodeCommit non utilizza ruoli collegati ai servizi.

Ruoli dei servizi

Questa funzionalità consente a un servizio di assumere un [ruolo di servizio](#) per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

CodeCommit non utilizza ruoli di servizio.

AWS CodeCommit esempi di politiche basate sull'identità

Per impostazione predefinita, gli utenti e i ruoli IAM non dispongono dell'autorizzazione per creare o modificare risorse CodeCommit. Inoltre, non possono eseguire attività utilizzando l'API Console di gestione AWS, AWS CLI, o AWS Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi collegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per esempi di policy, consultare i seguenti argomenti:

- [Esempio 1: consentire a un utente di eseguire operazioni in un'unica soluzione CodeCommit Regione AWS](#)
- [Esempio 2: consentire a un utente di utilizzare Git per un singolo repository](#)
- [Esempio 3: consentire a un utente che si connette da un intervallo di indirizzi IP specificato di accedere a un repository](#)
- [Esempio 4: negare o consentire azioni sulle filiali](#)
- [Esempio 5: negare o consentire azioni sui repository con tag](#)
- [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#)

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy nella scheda JSON](#) nella Guida per l'utente IAM.

Argomenti

- [Best practice delle policy](#)

- [Utilizzo della console di CodeCommit](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Visualizzazione basata sui tag CodeCommit repositories](#)

Best practice delle policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare CodeCommit risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consultare [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Utilizzo della console di CodeCommit

Per accedere alla AWS CodeCommit console, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle CodeCommit risorse nel tuo account Amazon Web Services. Se crei una policy basata su identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Per garantire che tali entità possano ancora utilizzare la CodeCommit console, allega anche la seguente politica AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente di IAM:

Per ulteriori informazioni, consulta [Utilizzo di policy basate sull'identità \(IAM Policies\) per CodeCommit](#).

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, è possibile accedere solo alle operazioni che soddisfano l'operazione API che stai cercando di eseguire.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"
    }
]
}

```

Visualizzazione basata sui tag CodeCommit *repositories*

È possibile utilizzare le condizioni nella policy basata sulle identità per controllare l'accesso alle risorse di CodeCommit in base ai tag. Per un criterio di esempio che illustra come eseguire questa operazione, consulta [Esempio 5: negare o consentire azioni sui repository con tag](#).

Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

Risoluzione dei problemi relativi AWS CodeCommit all'identità e all'accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con un CodeCommit IAM.

Argomenti

- [Non sono autorizzato a eseguire alcuna azione in CodeCommit](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Desidero visualizzare le mie chiavi di accesso](#)
- [Sono un amministratore e voglio consentire ad altri di accedere CodeCommit](#)
- [Desidero consentire a persone esterne al mio account Amazon Web Services di accedere alle mie CodeCommit risorse](#)

Non sono autorizzato a eseguire alcuna azione in CodeCommit

Se ti Console di gestione AWS dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Per ulteriori informazioni, consulta [Autorizzazioni necessarie per l'uso della console CodeCommit](#)

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a CodeCommit.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in CodeCommit. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Desidero visualizzare le mie chiavi di accesso

Dopo aver creato le chiavi di accesso utente IAM, è possibile visualizzare il proprio ID chiave di accesso in qualsiasi momento. Tuttavia, non è possibile visualizzare nuovamente la chiave di accesso segreta. Se perdi la chiave segreta, dovrai creare una nuova coppia di chiavi di accesso.

Le chiavi di accesso sono composte da due parti: un ID chiave di accesso (ad esempio AKIAIOSFODNN7EXAMPLE) e una chiave di accesso segreta (ad esempio, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). Come un nome utente e una password, è necessario utilizzare sia l'ID chiave di accesso sia la chiave di accesso segreta insieme per autenticare le richieste dell'utente. Gestisci le tue chiavi di accesso in modo sicuro mentre crei il nome utente e la password.

Important

Non fornire le chiavi di accesso a terze parti, neppure per aiutare a [trovare l'ID utente canonico](#). In questo modo, potresti concedere a qualcuno l'accesso permanente al tuo Account AWS.

Quando crei una coppia di chiavi di accesso, ti viene chiesto di salvare l'ID chiave di accesso e la chiave di accesso segreta in una posizione sicura. La chiave di accesso segreta è disponibile solo al momento della creazione. Se si perde la chiave di accesso segreta, è necessario aggiungere nuove chiavi di accesso all'utente IAM. È possibile avere massimo due chiavi di accesso. Se se ne hanno già due, è necessario eliminare una coppia di chiavi prima di crearne una nuova. Per visualizzare le istruzioni, consulta [Gestione delle chiavi di accesso](#) nella Guida per l'utente di IAM.

Sono un amministratore e voglio consentire ad altri di accedere CodeCommit

Per consentire ad altri di accedere CodeCommit, devi concedere l'autorizzazione alle persone o alle applicazioni che necessitano dell'accesso. Se si utilizza AWS IAM Identity Center per gestire persone e applicazioni, si assegnano set di autorizzazioni a utenti o gruppi per definirne il livello di accesso. I set di autorizzazioni creano e assegnano automaticamente le policy IAM ai ruoli IAM associati alla persona o all'applicazione. Per ulteriori informazioni, consulta [Set di autorizzazioni](#) nella Guida per l'AWS IAM Identity Center utente.

Se non utilizzi IAM Identity Center, devi creare entità IAM (utenti o ruoli) per le persone o le applicazioni che necessitano di accesso. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in CodeCommit. Dopo aver concesso le autorizzazioni, fornisci le credenziali

all'utente o allo sviluppatore dell'applicazione. Utilizzeranno tali credenziali per accedere. AWS Per ulteriori informazioni sulla creazione di utenti, gruppi, policy e autorizzazioni IAM, consulta [IAM Identities](#) and [Policies and permissions in IAM nella IAM User Guide](#).

Desidero consentire a persone esterne al mio account Amazon Web Services di accedere alle mie CodeCommit risorse

Per ulteriori informazioni, consulta [Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli](#).

Resilienza in AWS CodeCommit

L'infrastruttura AWS globale è costruita attorno Regioni AWS a zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Un CodeCommit repository o un modello di regola di CodeCommit approvazione esiste nel luogo in cui è stato creato. Regione AWS Per ulteriori informazioni, consulta [Regioni ed endpoint di connessione Git per AWS CodeCommit](#). Per garantire la resilienza nei repository, puoi configurare il tuo client Git in modo che esegua il push su due repository contemporaneamente. Per ulteriori informazioni, consulta [Invia i commit a un repository Git aggiuntivo](#).

[Per ulteriori informazioni sulle Regioni AWS zone di disponibilità, consulta AWS Global Infrastructure.](#)

Sicurezza dell'infrastruttura in AWS CodeCommit

In quanto servizio gestito, AWS CodeCommit è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper [Amazon Web Services: Overview of Security Processes](#).

Utilizzi chiamate API AWS pubblicate per accedere CodeCommit attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta che è associata a un'entità IAM. O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi chiamare queste operazioni API da qualsiasi posizione di rete, ma CodeCommit supporta restrizioni basate sull'indirizzo IP di origine. Puoi anche utilizzare CodeCommit le policy per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) specifici o specifici. VPCs In effetti, questo isola l'accesso alla rete a una determinata CodeCommit risorsa solo dal VPC specifico della rete. AWS

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Esempio 1: consentire a un utente di eseguire operazioni in un'unica soluzione CodeCommit Regione AWS](#)
- [Esempio 3: consentire a un utente che si connette da un intervallo di indirizzi IP specificato di accedere a un repository](#)
- [Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia](#)

Monitoraggio AWS CodeCommit

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle CodeCommit e altre AWS soluzioni esistenti. AWS fornisce i seguenti strumenti di monitoraggio per osservare CodeCommit, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon EventBridge può essere utilizzato per automatizzare AWS i tuoi servizi e rispondere automaticamente agli eventi di sistema, come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Per ulteriori informazioni, consulta [Amazon EventBridge User Guide](#) e [Monitoraggio CodeCommit degli eventi in Amazon EventBridge e Amazon CloudWatch Events](#).
- Amazon CloudWatch Events offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. CloudWatch Events consente l'elaborazione automatizzata basata sugli eventi, poiché puoi scrivere regole che controllano determinati eventi e attivano azioni automatizzate in altri AWS servizi quando si verificano tali eventi. Per ulteriori informazioni, consulta la [Amazon CloudWatch Events User Guide](#) e [Monitoraggio CodeCommit degli eventi in Amazon EventBridge e Amazon CloudWatch Events](#).
- Amazon CloudWatch Logs può essere utilizzato per monitorare, archiviare e accedere ai file di registro da CloudTrail e altre fonti. CloudWatch Logs possono monitorare le informazioni contenute nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account Amazon Web Services e invia i file di registro a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consulta la Guida per l'[AWS CloudTrail utente](#) e [Registrazione delle chiamate AWS CodeCommit API con AWS CloudTrail](#).

Monitoraggio CodeCommit degli eventi in Amazon EventBridge e Amazon CloudWatch Events

Puoi monitorare AWS CodeCommit gli eventi in EventBridge, il che fornisce un flusso di dati in tempo reale dalle tue applicazioni, applicazioni software-as-a-service (SaaS) e AWS servizi. EventBridge indirizza tali dati verso obiettivi come Amazon AWS Lambda Simple Notification Service. Questi eventi sono gli stessi che compaiono in Amazon CloudWatch Events, che fornisce un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle AWS risorse.

Gli esempi seguenti mostrano eventi per CodeCommit.

Note

CodeCommit supporta la fornitura `displayName` e l'inclusione di `emailAddress` informazioni nei tag di sessione negli eventi, se tali informazioni sono disponibili. Per ulteriori informazioni, consultare la sezione relativa ai [tag di sessione](#) e [Utilizzo dei tag per fornire informazioni sull'identità in CodeCommit](#).

Argomenti

- [evento referenceCreated](#)
- [evento referenceUpdated](#)
- [evento referenceDeleted](#)
- [unreferencedMergeCommitEvento creato](#)
- [commentOnCommitEvento creato](#)
- [commentOnCommitEvento aggiornato](#)
- [commentOnPullRequestCreated evento](#)
- [commentOnPullRequestUpdated evento](#)
- [pullRequestCreated evento](#)
- [pullRequestSourceBranchUpdated evento](#)
- [pullRequestStatusEvento modificato](#)
- [pullRequestMergeStatusUpdated evento](#)
- [approvalRuleTemplateEvento creato](#)

- [approvalRuleTemplateEvento aggiornato](#)
- [approvalRuleTemplateEvento eliminato](#)
- [approvalRuleTemplateAssociatedWithRepository evento](#)
- [approvalRuleTemplateDisassociatedWithRepository evento](#)
- [approvalRuleTemplateBatchAssociatedWithRepositories evento](#)
- [approvalRuleTemplateBatchDisassociatedFromRepositories evento](#)
- [pullRequestApprovalRuleCreated evento](#)
- [pullRequestApprovalRuleDeleted evento](#)
- [pullRequestApprovalRuleOverridden evento](#)
- [pullRequestApprovalStateChanged evento](#)
- [pullRequestApprovalRuleUpdated evento](#)
- [Evento ReactionCreated](#)
- [Evento ReactionUpdated](#)

evento referenceCreated

In questo evento di esempio, un ramo denominato myBranch è stato creato in un repository denominato 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

In questo evento di esempio, un ramo denominato myBranch è stato aggiornato mediante un'unione in un repository denominato 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

In questo evento di esempio, un ramo denominato myBranch è stato eliminato in un repository denominato 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"
  }
}
```

unreferencedMergeCommitEvento creato

In questo evento di esempio, è stato creato un commit di unione senza riferimenti in un repository denominato 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"
  }
}

```

commentOnCommitEvento creato

In questo evento di esempio, un utente federato denominato Mary_Major ha commentato un commit. In questo esempio, il provider di identità federata ha configurato i tag di sessione per `displayName` e `emailAddress`. Tali informazioni sono incluse nell'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"
  }
}

```

commentOnCommitEvento aggiornato

In questo evento di esempio, un utente che ha assunto un ruolo denominato Admin con un nome di sessione Mary_Major ha modificato un commento su un commit. In questo esempio, il ruolo includeva tag di sessione configurati per displayName e emailAddress. Tali informazioni sono incluse nell'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"
  }
}
```

commentOnPullRequestCreated evento

In questo evento di esempio, un utente federato denominato Saanvi_Sarkar ha commentato una richiesta pull. In questo esempio, il provider di identità federata ha configurato i tag di sessione per `displayName` e `emailAddress`. Tali informazioni sono incluse nell'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"
  }
}
```

commentOnPullRequestUpdated evento

In questo evento di esempio, un utente federato denominato `Saanvi_Sarkar` ha modificato un commento su una richiesta pull. In questo esempio, il provider di identità federata ha configurato i tag di sessione per `displayName` e `emailAddress`. Tali informazioni sono incluse nell'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"
  }
}
```

pullRequestCreated evento

In questo evento di esempio, è stata creata una richiesta pull in un repository denominato `MyDemoRepo` da un utente che ha assunto un ruolo denominato `Admin` con un nome di sessione di

Mary_Major. Non sono state fornite informazioni sui tag di sessione, pertanto le informazioni non sono incluse nell'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"
  }
}
```

pullRequestSourceBranchUpdated evento

In questo evento di esempio, un utente che ha assunto un ruolo denominato Admin con un nome di sessione Mary_Major ha aggiornato il ramo di origine denominato test-branch per una richiesta pull con l'ID 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"
  }
}
```

```
}
```

pullRequestStatusEvento modificato

In questo evento di esempio, un utente che ha assunto un ruolo denominato Admin con un nome di sessione Mary_Major ha chiuso una richiesta pull con l'ID 1. La richiesta pull non è stata unita.

```
{
  "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"
  }
}
```

```
}
```

pullRequestMergeStatusUpdated evento

In questo evento di esempio, un utente che ha assunto un ruolo denominato Admin con un nome di sessione di Mary_Major ha eseguito l'unione di una richiesta pull con l'ID di 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"
  }
}
```

```
}
```

approvalRuleTemplateEvento creato

In questo evento di esempio, un utente con un nome utente IAM Mary_Major ha creato un modello di regola di approvazione denominato 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": {}
  }
}
```

approvalRuleTemplateEvento aggiornato

In questo evento di esempio, un utente con un nome utente IAM ha Mary_Major modificato un modello di regola di approvazione denominato 2-approvers-required-for-main. Il modello della regola di approvazione non è associato ad alcun repository.

```
{
  "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": {}
}
}

```

approvalRuleTemplateEvento eliminato

In questo evento di esempio, un utente con un nome utente IAM ha Mary_Major eliminato un modello di regola di approvazione denominato 2-approvers-required-for-main. Il modello della regola di approvazione non è associato ad alcun repository.

```

{
  "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": {}
  }
}

```

approvalRuleTemplateAssociatedWithRepository evento

In questo evento di esempio, un utente con un nome utente IAM ha Mary_Major associato un modello di regola di approvazione denominato 2-approvers-required-for-main un repository denominato 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"
  }
}
}

```

approvalRuleTemplateDisassociatedWithRepository evento

In questo evento di esempio, un utente con un nome utente IAM ha Mary_Major dissociato un modello di regola di approvazione denominato 2-approvers-required-for-main da un repository denominato. 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"
    }
  }
}

```

```
}
```

approvalRuleTemplateBatchAssociatedWithRepositories evento

In questo evento di esempio, un utente con un nome utente IAM di Mary_Major batch ha associato un modello di regola di approvazione denominato 2-approvers-required-for-main a un repository denominato MyDemoRepo e a un repository denominato. 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"
    }
  }
}
```

approvalRuleTemplateBatchDisassociatedFromRepositories evento

In questo evento di esempio, un utente con un nome utente IAM Mary_Major batch ha dissociato un modello di regola di approvazione denominato 2-approvers-required-for-main da un repository denominato MyDemoRepo e un repository denominato. 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"
    }
  }
}
```

pullRequestApprovalRuleCreated evento

In questo evento di esempio, un utente con un nome utente IAM di Mary_Major ha creato una regola di approvazione denominata 1-approver-needed per una pull request con l'ID di227.

```
{
```

```

"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"
}
}

```

pullRequestApprovalRuleDeleted evento

In questo evento di esempio, un utente con un nome utente IAM di ha Mary_Major eliminato una regola di approvazione denominata 1-approver-needed per una pull request con l'ID di227. Un utente IAM con lo stesso nome ha Saanvi_Sarkar originariamente creato la regola di approvazione.

```
{
  "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"
  }
}

```

pullRequestApprovalRuleOverridden evento

In questo evento di esempio, i requisiti della regola di approvazione per una pull request sono stati accantonati (OVERRIDE) da un utente con un nome utente IAM di `Mary_Major`. La pull request è stata creata da un utente con un nome utente IAM di `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"
}

```

In questo caso di esempio, i requisiti della regola di approvazione per una richiesta pull sono stati ripristinati (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"
}
}

```

pullRequestApprovalStateChanged evento

In questo evento di esempio, una pull request è stata approvata da un utente con un nome utente IAM di Mary_Major.

```

{
    "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"
}
}

```

In questo evento di esempio, l'approvazione per una pull request è stata revocata da un utente con un nome utente IAM di `Mary_Major`.

```

{
    "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"
}
}

```

pullRequestApprovalRuleUpdated evento

In questo evento di esempio, una regola di approvazione per una pull request è stata modificata da un utente con un nome utente IAM di `Mary_Major`. È anche l'utente che ha creato la richiesta pull.

```

{
    "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

In questo evento di esempio, una reazione a un commento è stata aggiunta da un utente con un nome utente IAM di 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

In questo evento di esempio, una reazione a un commento è stata aggiornata da un utente con un nome utente IAM di `Mary_Major`. Gli utenti possono solo aggiornare le proprie reazioni.

```

{
  "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"
}

```

```
}
```

Registrazione delle chiamate AWS CodeCommit API con AWS CloudTrail

CodeCommit è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in CodeCommit. CloudTrail acquisisce tutte le chiamate API per eventi CodeCommit, incluse le chiamate dalla CodeCommit console, dal client Git e dalle chiamate in codice. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per CodeCommit. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata CodeCommit, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la [Guida AWS CloudTrail per l'utente](#).

CodeCommit informazioni in CloudTrail

CloudTrail è abilitato sul tuo account Amazon Web Services al momento della creazione dell'account. Quando si verifica un'attività in CodeCommit, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nell'account Amazon Web Services. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo account Amazon Web Services, inclusi gli eventi per CodeCommit, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- [Panoramica della creazione di un trail](#)
- [CloudTrail Servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Quando CloudTrail la registrazione è abilitata nel tuo account Amazon Web Services, le chiamate API effettuate alle CodeCommit azioni vengono tracciate nei file di CloudTrail registro, dove vengono scritte insieme ad altri record di AWS servizio. CloudTrail determina quando creare e scrivere su un nuovo file in base a un periodo di tempo e alle dimensioni del file.

Tutte CodeCommit le azioni vengono registrate da CloudTrail, incluse alcune (ad esempio `GetObjectIdentifier`) che non sono attualmente documentate nell'[AWS CodeCommit API Reference](#), ma sono invece indicate come autorizzazioni di accesso e sono documentate in. [Riferimento per le autorizzazioni CodeCommit](#) Ad esempio, le chiamate alle azioni `ListRepositories` (in AWS CLI, `aws codecommit list-repositories`), `CreateRepository` (`aws codecommit create-repository`) e `PutRepositoryTriggers` (`aws codecommit put-repository-triggers`) generano voci nei file di CloudTrail registro, così come le chiamate del client Git a `GitPull` and `GitPush`. Inoltre, se avete un CodeCommit repository configurato come sorgente per una pipeline in CodePipeline, vedrete le chiamate alle azioni di autorizzazione di CodeCommit accesso come `UploadArchive from. CodePipeline` Poiché CodeCommit viene utilizzato AWS Key Management Service per crittografare e decrittografare gli archivi, nei log verranno visualizzate anche le chiamate da CodeCommit a `Encrypt` e le azioni provenienti da. `Decrypt AWS KMS CloudTrail`

Ogni voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni sull'identità dell'utente nella voce di log ti permettono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente IAM o root
- Se la richiesta è stata effettuata con credenziali di sicurezza temporanee per un ruolo o un utente federato oppure da un altro servizio.
- Se la richiesta è stata effettuata da un altro servizio AWS

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Puoi archiviare i file di log nel tuo bucket Amazon S3 per tutto il tempo che desideri, ma puoi anche definire regole del ciclo di vita di Amazon S3 per archiviare o eliminare automaticamente i file di registro. Per impostazione predefinita, i file di log sono crittografati mediante la crittografia lato server (SSE) di Amazon S3.

Comprendere CodeCommit le voci dei file di registro

CloudTrail i file di registro possono contenere una o più voci di registro. Ogni voce elenca più eventi in formato JSON. Un evento di log rappresenta una singola richiesta inviata da un'origine e include

informazioni sull'operazione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. Le voci di log non sono una traccia di stack ordinata delle chiamate API pubbliche, pertanto non vengono visualizzate in un ordine specifico.

Note

Questo esempio è stato formattato per migliorare la leggibilità. In un file di CloudTrail registro, tutte le voci e gli eventi sono concatenati in un'unica riga. Anche questo esempio è stato limitato a una singola voce. CodeCommit In un vero file di CloudTrail registro, vengono visualizzate voci ed eventi provenienti da più AWS servizi.

Indice

- [Esempio: una voce di registro per elencare gli CodeCommit archivi](#)
- [Esempio: una voce di registro per la creazione di un repository CodeCommit](#)
- [Esempi: voci di log per le chiamate pull Git a un repository CodeCommit](#)
- [Esempio: una voce di registro per un invio riuscito a un repository CodeCommit](#)

Esempio: una voce di registro per elencare gli CodeCommit archivi

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`ListRepositories` azione.

Note

Sebbene `ListRepositories` restituisca un elenco di repository, le risposte non modificabili non vengono registrate nei CloudTrail log, quindi vengono mostrate come `null` nel `responseElements` file di 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"
  }
}
```

Esempio: una voce di registro per la creazione di un repository CodeCommit

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'CreateRepositoryazione nella regione degli Stati Uniti orientali (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 botocore/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"
}

```

Esempi: voci di log per le chiamate pull Git a un repository CodeCommit

L'esempio seguente mostra una voce di CloudTrail registro che mostra l'GitPullazione in cui è già presente il repository locale. up-to-date

```

{
  "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"
  }
}

```

L'esempio seguente mostra una voce di CloudTrail registro che mostra l'GitPullazione laddove il repository locale non lo è up-to-date e quindi i dati vengono trasferiti dal CodeCommit repository al repository locale.

```

{
  "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"
}

```

Esempio: una voce di registro per un invio riuscito a un repository CodeCommit

L'esempio seguente mostra una voce di CloudTrail registro che dimostra un'azione riuscita `GitPush`. Se il push riesce, l'operazione `GitPush` viene visualizzata due volte in una voce di log.

```

{
  "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"
}
```

Creazione di CodeCommit risorse con AWS CloudFormation

AWS CodeCommit è integrato con AWS CloudFormation, un servizio che consente di modellare e configurare le AWS risorse in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Crei un modello che descrive tutte le AWS risorse che desideri (ad esempio gli archivi) e fornisce e CloudFormation configura tali risorse per te.

Quando lo utilizzi CloudFormation, puoi riutilizzare il modello per configurare le CodeCommit risorse in modo coerente e ripetuto. Descrivi le tue risorse una sola volta, quindi fornisci le stesse risorse più e più volte in più Account AWS aree geografiche.

CodeCommit e CloudFormation modelli

Per fornire e configurare le risorse CodeCommit e i servizi correlati, è necessario conoscere [CloudFormation i modelli](#). I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri fornire negli CloudFormation stack. Se non conosci JSON o YAML, puoi usare CloudFormation Designer per iniziare a usare i modelli. CloudFormation Per ulteriori informazioni, consulta [Che cos'è CloudFormation Designer?](#) nella Guida per l'utente di AWS CloudFormation .

CodeCommit supporta la creazione di repository in CloudFormation A differenza della creazione di repository dalla console o dalla riga di comando, è possibile utilizzare CloudFormation per creare repository e inviare automaticamente il codice nel repository appena creato da un file.zip specificato in un bucket Amazon S3. Per ulteriori informazioni, inclusi esempi di modelli JSON e YAML per i repository, consulta. [AWS::CodeCommit::Repository](#)

Quando crei un CodeCommit repository utilizzando CloudFormation, hai la possibilità di eseguire il commit del codice in quel repository come parte del processo di creazione, purché l'archivio sia inferiore a 20 MB, configurando le proprietà in [AWS::CodeCommit::Repository](#) Code. Puoi specificare il bucket Amazon S3 in cui è archiviato il codice e, facoltativamente, utilizzare la [BranchName proprietà](#) per specificare il nome del ramo predefinito che verrà creato nel commit iniziale di quel codice. Queste proprietà vengono utilizzate solo nella creazione iniziale del repository e vengono ignorate negli aggiornamenti dello stack. Non è possibile utilizzare queste proprietà per effettuare ulteriori commit su un repository o per modificare il nome del ramo predefinito dopo il commit iniziale.

Note

Il 19 gennaio 2021, AWS ha cambiato il nome del ramo predefinito CodeCommit da master a main. Questa modifica del nome influisce sul comportamento predefinito di CodeCommit quando si crea il commit iniziale per i repository utilizzando la CodeCommit console, le CodeCommit APIs, gli AWS SDKs, e il AWS CLI. I repository creati con CloudFormation o AWS CDK con un commit iniziale di codice come parte della creazione si allineano a questa modifica a partire dal 4 marzo 2021. Questa modifica non influisce sugli archivi o sulle filiali esistenti. I clienti che utilizzano client Git locali per creare i propri commit iniziali hanno un nome di ramo predefinito che segue la configurazione di tali client Git. Per ulteriori informazioni, consulta [Lavorare con le filiali](#), [Creare un commit](#) e [Modificare le impostazioni delle filiali](#).

Puoi anche creare modelli per creare risorse correlate, come [regole di notifica](#) per gli archivi, [AWS CodeBuild creare progetti](#), [AWS CodeDeploy applicazioni](#) e [AWS CodePipeline pipeline](#).

Esempi di modello

Gli esempi seguenti creano un CodeCommit repository denominato *MyDemoRepo*. Il repository appena creato è popolato con codice memorizzato in un bucket Amazon S3 *MySourceCodeBucket* denominato e inserito in un ramo *development* denominato, che è il ramo predefinito per il repository.

Note

Il nome del bucket Amazon S3 che contiene il file ZIP con il contenuto che verrà salvato nel nuovo repository può essere specificato utilizzando un ARN o il nome del bucket nell'account Amazon Web Services. La chiave oggetto Amazon S3 è definita nella [Amazon S3 Developer Guide](#).

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
```

Per ulteriori esempi, consulta [AWS::CodeCommit::Repository](#).

CloudFormation, CodeCommit, e AWS Cloud Development Kit (AWS CDK)

Repository creati utilizzando la CloudFormation funzionalità AWS CDK use durante la loro creazione. Capire come CloudFormation i modelli interagiscono con CodeCommit le risorse può aiutarti a creare e gestire il AWS CDK codice. Per ulteriori informazioni su AWS CDK, consulta la [Guida per gli AWS Cloud Development Kit \(AWS CDK\) sviluppatori](#) e l'[AWS CDK API Reference](#).

Il seguente esempio di AWS CDK Typescript crea un CodeCommit repository denominato.

MyDemoRepo Il repository appena creato è popolato con codice memorizzato in un bucket Amazon S3 *MySourceCodeBucket* denominato e inserito in un ramo *development* denominato, che è il ramo predefinito per il repository.

```
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"
        }
      },
    });
  }
}
```

Scopri di più su CloudFormation

Per ulteriori informazioni CloudFormation, consulta le seguenti risorse:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guida per l'utente](#)
- [AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando](#)

Risoluzione dei problemi AWS CodeCommit

Le informazioni seguenti possono risultare utili per risolvere i problemi comuni di AWS CodeCommit.

Argomenti

- [Risoluzione dei problemi relativi alle credenziali Git e alle connessioni HTTPS a AWS CodeCommit](#)
- [Risoluzione dei problemi git-remote-codecommit e AWS CodeCommit](#)
- [Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit](#)
- [Risoluzione dei problemi relativi all'helper delle credenziali e alle connessioni HTTPS a AWS CodeCommit](#)
- [Risoluzione dei problemi relativi ai client Git e AWS CodeCommit](#)
- [Risoluzione degli errori di accesso e AWS CodeCommit](#)
- [Risoluzione degli errori di configurazione e AWS CodeCommit](#)
- [Risoluzione degli errori della console e AWS CodeCommit](#)
- [Risoluzione dei problemi relativi ai trigger e AWS CodeCommit](#)
- [Attiva il debug](#)

Risoluzione dei problemi relativi alle credenziali Git e alle connessioni HTTPS a AWS CodeCommit

Le seguenti informazioni potrebbero aiutarti a risolvere i problemi più comuni relativi all'utilizzo di credenziali Git e HTTPS per connetterti ai repository. AWS CodeCommit

Argomenti

- [Credenziali Git per AWS CodeCommit: Continuo a visualizzare una richiesta di credenziali quando mi connetto al mio CodeCommit repository dal terminale o dalla riga di comando](#)
- [Credenziali Git per AWS CodeCommit: Ho impostato le credenziali Git, ma il mio sistema non le utilizza](#)

Credenziali Git per AWS CodeCommit: Continuo a visualizzare una richiesta di credenziali quando mi connetto al mio CodeCommit repository dal terminale o dalla riga di comando

Problema: quando provi a inviare, estrarre o interagire in altro modo con un CodeCommit repository dal terminale o dalla riga di comando, ti viene richiesto di fornire un nome utente e una password e devi fornire le credenziali Git per il tuo utente IAM.

Possibili correzioni: Le cause più comuni di questo errore sono che il computer locale esegue un sistema operativo che non supporta la gestione delle credenziali, o non ha un'utilità di gestione delle credenziali installata, oppure le credenziali Git per il tuo utente IAM non sono state salvate in uno di questi sistemi di gestione delle credenziali. A seconda del sistema operativo in uso e dell'ambiente locale, può essere necessario installare un sistema di gestione delle credenziali, configurare quello incluso nel sistema operativo oppure personalizzare l'ambiente locale per l'utilizzo di un sistema di storage delle credenziali. Ad esempio, se sul computer è in esecuzione macOS, è possibile utilizzare l'utilità Keychain Access per archiviare le credenziali. Se il computer esegue Windows puoi utilizzare la funzionalità Git Credential Manager installata con Git per Windows. Per ulteriori informazioni, consulta [Per utenti HTTPS che utilizzano credenziali Git](#) e [Credential Storage](#) nella documentazione di Git.

Credenziali Git per AWS CodeCommit: Ho impostato le credenziali Git, ma il mio sistema non le utilizza

Problema: quando provi a utilizzare CodeCommit con un client Git, sembra che il client non utilizzi le credenziali Git per il tuo utente IAM.

Possibili soluzioni. La causa più comune di questo errore è la precedente configurazione del computer per l'utilizzo dell'assistente credenziali incluso in AWS CLI. Verificare la presenza di sezioni di configurazione simili alla seguente nel file .gitconfig e rimuoverle:

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

Salvare il file e quindi aprire una nuova riga di comando o sessione del terminale prima di riprovare la connessione.

È anche possibile che nel computer siano configurati più assistenti credenziali o sistemi di gestione delle credenziali e che il sistema utilizzi, per impostazione predefinita, un'altra configurazione. Per reimpostare l'assistente credenziali utilizzato per impostazione predefinita, è possibile utilizzare l'opzione `--system` anziché `--global` o `--local` quando si esegue il comando `git config`.

Per ulteriori informazioni, consulta [Per utenti HTTPS che utilizzano credenziali Git](#) e [Credential Storage](#) nella documentazione di Git.

Risoluzione dei problemi git-remote-codecommit e AWS CodeCommit

Le informazioni seguenti possono essere utili per risolvere gli errori relativi a git-remote-codecommit durante la connessione ai repository AWS CodeCommit .

Argomenti

- [Vedo un errore: git: 'remote-codecommit' non è un comando git](#)
- [Vedo un errore: fatale: impossibile trovare l'helper remoto per 'codecommit'](#)
- [Errore di clonazione: non riesco a clonare un CodeCommit repository da un IDE](#)
- [Errore push o pull: impossibile eseguire il push o il pull dei commit da un IDE in un repository CodeCommit](#)
- [Errore di installazione: viene visualizzato un externally-managed-environment errore durante l'installazione git-remote-codecommit](#)

Vedo un errore: git: 'remote-codecommit' non è un comando git

Problema: quando si tenta di utilizzare git-remote-codecommit, viene visualizzato un errore che non git-remote-codecommit è un comando git. Vedi 'git --help'».

Possibili correzioni: Il motivo più comune di questo errore è che non hai aggiunto l' `git-remote-codecommit` eseguibile al tuo `PATH` o che la stringa contiene un errore di sintassi. Questo può accadere quando manca un trattino tra `git` e `remote-codecommit` o quando prima viene inserito un `git` aggiuntivo. `git-remote-codecommit`

Potresti riscontrare questo errore anche se hai aggiornato l'installazione locale a una versione non supportata. Per ulteriori informazioni sulle versioni supportate di Python, consulta. [git-remote-codecommit](#)

Per ulteriori informazioni sulla configurazione e l'utilizzo git-remote-codecommit, vedere [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Vedo un errore: fatale: impossibile trovare l'helper remoto per 'codecommit'

Problema: quando si tenta di utilizzare git-remote-codecommit, viene visualizzato un errore che indica «fatale: impossibile trovare l'helper remoto per 'codecommit'».

Possibili correzioni: i motivi più comuni di questo errore sono:

- La configurazione non è completa per git-remote-codecommit
- L'installazione è stata git-remote-codecommit eseguita in una posizione non presente nel percorso dell'utente o non è configurata come parte della variabile di Path ambiente
- Python non è nel tuo percorso o non è configurato come parte della variabile di ambiente Path
- Stai utilizzando un terminale o una finestra della riga di comando che non è stata riavviata dopo il completamento dell'installazione git-remote-codecommit

Per ulteriori informazioni sulla configurazione e l'utilizzo git-remote-codecommit, vedere. [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#)

Errore di clonazione: non riesco a clonare un CodeCommit repository da un IDE

Problema: quando si tenta di clonare un CodeCommit repository in un IDE, viene visualizzato un errore che indica che l'endpoint o l'URL non sono validi.

Possibili correzioni: non tutte IDEs supportano l'URL utilizzato da durante la clonazione. git-remote-codecommit Clona il repository localmente dal terminale o dalla riga di comando e quindi aggiungi quel repository locale all'IDE. Per ulteriori informazioni, consulta [Passaggio 3: Connect alla CodeCommit console e clona il repository](#).

Errore push o pull: impossibile eseguire il push o il pull dei commit da un IDE in un repository CodeCommit

Problema: quando tenti di eseguire il pull o il push da un IDE, viene visualizzato un errore di connessione.

Possibili soluzioni: il motivo più comune di questo errore è che l'IDE non è compatibile con helper remoti Git come `git-remote-codecommit`. Invece di utilizzare la funzionalità IDE per eseguire il commit, il push e il pull del codice, aggiorna manualmente il repository locale dalla riga di comando o dal terminale utilizzando i comandi Git.

Per ulteriori informazioni sugli helper remoti e Git, consulta la [documentazione di Git](#).

Errore di installazione: viene visualizzato un externally-managed-environment errore durante l'installazione git-remote-codecommit

Problema: quando si tenta di eseguire il `pip install git-remote-codecommit` comando, viene visualizzato un errore che indica che l'ambiente è gestito esternamente.

Possibili correzioni: Il motivo più comune di questo errore è che stai eseguendo una distribuzione (distro) di Python che definisce un file marker GESTITO ESTERNAMENTE. La soluzione migliore per questo è creare e utilizzare un ambiente virtuale.

Per ulteriori informazioni sugli ambienti gestiti esternamente e Python, vedere Ambienti [gestiti esternamente](#) e [Installare pacchetti in un ambiente virtuale usando pip e venv nella](#) documentazione di Python.

Risoluzione dei problemi relativi alle connessioni SSH a AWS CodeCommit

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi all'utilizzo di SSH per la connessione ai repository. CodeCommit

Argomenti

- [Errore di accesso: la chiave pubblica viene caricata correttamente su IAM ma la connessione non riesce su sistemi Linux, macOS o Unix](#)
- [Errore di accesso: la chiave pubblica è stata caricata correttamente su IAM e SSH è stato testato correttamente, ma la connessione non riesce sui sistemi Windows](#)
- [Sfida di autenticazione: non è possibile stabilire l'autenticità dell'host durante la connessione a un repository CodeCommit](#)
- [Errore IAM: 'Formato non valido' quando si tenta di aggiungere una chiave pubblica a IAM](#)
- [Devo accedere ai CodeCommit repository in più account Amazon Web Services con credenziali SSH](#)

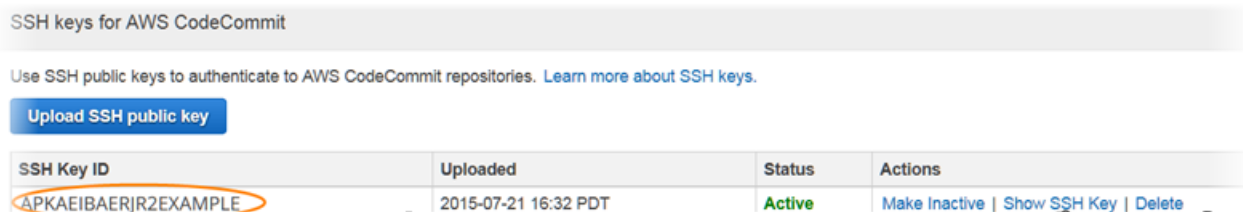
- [Git in Windows: l'emulatore Bash o la riga di comando si blocca durante il tentativo di connessione tramite SSH](#)
- [Il formato della chiave pubblica richiede una specifica in alcune distribuzioni di Linux](#)
- [Errore di accesso: chiave pubblica SSH negata durante la connessione a un repository CodeCommit](#)

Errore di accesso: la chiave pubblica viene caricata correttamente su IAM ma la connessione non riesce su sistemi Linux, macOS o Unix

Problema: quando si tenta di connettersi a un endpoint SSH per comunicare con un CodeCommit repository, durante il test della connessione o la clonazione di un repository, la connessione fallisce o viene rifiutata.

Possibili correzioni: l'ID della chiave SSH assegnato alla chiave pubblica in IAM potrebbe non essere associato al tentativo di connessione. [Potresti non aver configurato un file di configurazione](#), potresti non avere accesso al file di configurazione, un'altra impostazione potrebbe impedire la corretta lettura del file di configurazione, potresti aver fornito l'ID della chiave sbagliato o potresti aver fornito l'ID dell'utente IAM anziché l'ID della chiave.

L'ID della chiave SSH si trova nella console IAM nel profilo del tuo utente 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

Se hai IDs caricato più di una chiave SSH, le chiavi sono elencate alfabeticamente per ID chiave, non per data di caricamento. Assicurati di aver copiato l'ID della chiave associato alla data di caricamento corretta.

Provare a testare la connessione con il comando seguente:

```
ssh Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com
```

Se dopo aver confermato la connessione viene visualizzato un messaggio di operazione riuscita, l'ID della chiave SSH è valido. Modifica il file di configurazione per associare i tuoi tentativi di connessione alla tua chiave pubblica in IAM. Se per qualche motivo preferisci non modificare il file di configurazione, puoi anteporre l'ID della chiave SSH a tutti i tentativi di connessione al repository. Ad esempio, se desideri clonare un repository denominato *MyDemoRepo* senza modificare il file di configurazione per associare i tentativi di connessione, esegui il comando seguente:

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

Per ulteriori informazioni, consulta [Per connessioni SSH su Linux, macOS o Unix](#).

Errore di accesso: la chiave pubblica è stata caricata correttamente su IAM e SSH è stato testato correttamente, ma la connessione non riesce sui sistemi Windows

Problema: quando si tenta di utilizzare un endpoint SSH per clonare o comunicare con un CodeCommit repository, viene visualizzato un messaggio di errore contenente la frase. No supported authentication methods available

Possibili soluzioni. Il motivo più comune di questo errore è l'impostazione di una variabile di ambiente del sistema Windows che indica a Windows di usare un altro programma quando si tenta di usare SSH. Ad esempio, è possibile che sia stata impostata una variabile GIT_SSH che punta a uno dei set di strumenti PuTTY (plink.exe). Potrebbe trattarsi di una configurazione legacy oppure potrebbe essere una configurazione richiesta per uno o più programmi installati nel computer. Se sei certo che questa variabile di ambiente non è richiesta, puoi rimuoverla aprendo le proprietà di sistema.

Per risolvere il problema, aprire un emulatore Bash e quindi riprovare la connessione SSH, ma includere GIT_SSH_COMMAND="SSH" come prefisso. Ad esempio, per clonare un repository usando SSH:

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

Un problema simile può verificarsi se la versione di Windows richiede di includere l'ID della chiave SSH nella stringa di connessione al momento della connessione tramite SSH alla riga di comando di Windows. Riprova a connetterti, questa volta includendo l'ID della chiave SSH copiato da IAM come parte del comando. Esempio:

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

Sfida di autenticazione: non è possibile stabilire l'autenticità dell'host durante la connessione a un repository CodeCommit

Problema: quando si tenta di utilizzare un endpoint SSH per comunicare con un CodeCommit repository, viene visualizzato un messaggio di avviso contenente la frase The authenticity of host '*host-name*' can't be established.

Possibili soluzioni. È possibile che le credenziali non siano impostate correttamente. Segui le istruzioni fornite in [Per connessioni SSH su Linux, macOS o Unix](#) o [Per le connessioni SSH su Windows](#).

Se hai seguito questi passaggi e il problema persiste, è possibile che qualcuno stia tentando un attacco. man-in-the-middle Quando viene visualizzato il messaggio seguente, digitare no e premere Invio.

```
Are you sure you want to continue connecting (yes/no)?
```

Assicurati che l'impronta digitale e la chiave pubblica per CodeCommit le connessioni corrispondano a quelle documentate negli argomenti sulla configurazione SSH prima di continuare con la connessione.

Impronte digitali pubbliche per CodeCommit

Server	Tipo di hash crittografico	Impronta digitale
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 di hash crittografico	Impronta digitale
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 di hash crittografico	Impronta digitale
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 Qr1lTZA
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 di hash crittografico	Impronta digitale
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 di hash crittografico	Impronta digitale
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/U717nX/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 di hash crittografico	Impronta digitale
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
git-codecommit.ap-east-1.amazonaws.com	SHA256	LafadYwUYW8h0NoTRp objjNs9IRnbEwHtezD 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

Server	Tipo di hash crittografico	Impronta digitale
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	1yXrWbCg3uQmJr1l1Xx 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 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

Server	Tipo di hash crittografico	Impronta digitale
git-codecommit.il-central-1.amazonaws.com	SHA256	uFxhp51kUWh1eTLeYb xQVYm4RnNLNZ5Dbdm1 cgdS1/8

Errore IAM: 'Formato non valido' quando si tenta di aggiungere una chiave pubblica a IAM

Problema: in IAM, quando si tenta di configurare l'utilizzo di SSH con CodeCommit, viene visualizzato un messaggio di errore contenente la frase «Invalid formatquando si tenta di aggiungere la chiave pubblica».

Possibili correzioni: IAM richiede che la chiave pubblica sia codificata in formato ssh-rsa o in formato PEM. Accetta chiavi pubbliche solo nel formato OpenSSH e ha requisiti aggiuntivi come specificato [in Use SSH Keys CodeCommit with the IAM User Guide](#). Se la chiave pubblica viene specificata in un altro formato o se la chiave non contiene il numero richiesto di bit, viene visualizzato questo errore.

- Quando copi la chiave pubblica SSH, il sistema operativo potrebbe introdurre interruzioni di riga. Assicurati che non vi siano interruzioni di riga nella chiave pubblica che aggiungi a IAM.
- Alcuni sistemi operativi Windows non utilizzano il formato OpenSSH. Per generare una coppia di key pair e copiare il formato OpenSSH richiesto da IAM, consulta [the section called “Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit”](#)

Per ulteriori informazioni sui requisiti per le chiavi SSH in IAM, consulta [Use SSH Keys with CodeCommit nella IAM User Guide](#).

Devo accedere ai CodeCommit repository in più account Amazon Web Services con credenziali SSH

Problema: voglio configurare l'accesso SSH ai CodeCommit repository in più di un account Amazon Web Services.

Possibili correzioni: puoi creare coppie di public/private chiavi SSH uniche per ogni account Amazon Web Services e configurare IAM con ciascuna chiave. Puoi quindi configurare il tuo file `~/.ssh/config` con informazioni su ogni ID utente IAM associato alla chiave pubblica. Esempio:

```
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.
```

In questa configurazione, potrai sostituire 'git-codecommit.us-east-1.amazonaws.com' con 'codecommit-2'. Ad esempio, per clonare un repository nel tuo secondo account Amazon Web Services:

```
git clone ssh://codecommit-2/v1/repos/YourRepositoryName
```

Per configurare un telecomando per il tuo repository, esegui. git remote add Esempio:

```
git remote add origin ssh://codecommit-2/v1/repos/YourRepositoryName
```

Per altri esempi, vedi [questo post sul forum](#) e [questo contributo su GitHub](#).

Git in Windows: l'emulatore Bash o la riga di comando si blocca durante il tentativo di connessione tramite SSH

Problema. Dopo aver configurato l'accesso SSH per Windows e aver confermato la connettività alla riga di comando o al terminale, viene visualizzato un messaggio indicante che la chiave host del server non è memorizzata nella cache del Registro di sistema e il prompt per archiviare la chiave nella cache risulta bloccato (non accetta s/n né restituisce input) quando si tenta di utilizzare comandi quali git pull, git push o git clone al prompt dei comandi o nell'emulatore Bash.

Possibili soluzioni. La causa più comune di questo errore è la configurazione dell'ambiente Git per l'utilizzo di un sistema di autenticazione diverso da OpenSSH (probabilmente PuTTY). Ciò causa problemi con la memorizzazione delle chiavi nella cache in alcune configurazioni. Per risolvere questo problema, prova una delle soluzioni seguenti:

- Aprire un emulatore Bash e aggiungere il parametro `GIT_SSH_COMMAND="ssh"` prima del comando Git. Ad esempio, se si sta tentando di eseguire il push in un repository, anziché digitare `git push`, digitare:

```
GIT_SSH_COMMAND="ssh" git push
```

- Se hai installato PuTTY, apri PuTTY e in Nome host (o indirizzo IP), inserisci l'endpoint che desideri raggiungere (ad CodeCommit esempio, `git-codecommit.us-east-2.amazonaws.com`). Seleziona Apri. Quando richiesto dall'avviso di sicurezza di PuTTY, scegliere Yes (Sì) per memorizzare la chiave nella cache in modo permanente.
- Rinominare o eliminare la variabile di ambiente `GIT_SSH` se non la si utilizza più. Quindi aprire un nuovo prompt dei comandi o una nuova sessione dell'emulatore Bash e riprovare a eseguire il comando.

Per altre soluzioni, vedi [Git si blocca clone/pull continuamente nella chiave Store nella cache](#) su Stack Overflow.

Il formato della chiave pubblica richiede una specifica in alcune distribuzioni di Linux

Problema: quando si tenta di configurare una coppia di key pair pubblica-privata, viene visualizzato un errore.

Possibili correzioni: alcune distribuzioni di Linux richiedono una riga di configurazione aggiuntiva nel `~/.ssh/config` file che specifica i tipi di chiavi pubbliche accettati. Per ulteriori informazioni, consulta la documentazione relativa alla tua distribuzione. `PubkeyAcceptedKeyTypes`

Errore di accesso: chiave pubblica SSH negata durante la connessione a un repository CodeCommit

Problema: quando si tenta di utilizzare un endpoint SSH per comunicare con un CodeCommit repository, viene visualizzato un messaggio di errore contenente la frase. `Error: public key denied`

Possibili soluzioni: il motivo più comune di questo errore è il mancato completamento della configurazione delle connessioni SSH. Configura una coppia di chiavi SSH pubblica e privata, quindi associa la chiave pubblica al tuo utente IAM. Per ulteriori informazioni sulla configurazione di SSH, consulta [Per connessioni SSH su Linux, macOS o Unix](#) e [Per le connessioni SSH su Windows](#).

Risoluzione dei problemi relativi all'helper delle credenziali e alle connessioni HTTPS a AWS CodeCommit

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi all'utilizzo dell'helper per le credenziali incluso in AWS CLI e di HTTPS per connetterti ai repository. CodeCommit

Note

Sebbene l'helper per le credenziali sia un metodo supportato per la connessione CodeCommit tramite accesso federato, un provider di identità o credenziali temporanee, il metodo consigliato consiste nell'installare e utilizzare l'utilità. `git-remote-codecommit` Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#).

Argomenti

- [Ricevo un errore durante l'esecuzione del `git config` comando per configurare l'helper delle credenziali](#)
- [Vedo un errore durante la lettura del nome utente durante il tentativo di clonare un repository](#)
- [Ricevo un errore di comando non trovato in Windows durante l'utilizzo dell'assistente credenziali](#)
- [Mi viene richiesto un nome utente quando mi connetto a un repository CodeCommit](#)
- [Git per macOS: ho configurato correttamente l'helper delle credenziali, ma ora mi viene negato l'accesso al mio repository \(403\)](#)
- [Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository \(403\)](#)

Ricevo un errore durante l'esecuzione del **git config** comando per configurare l'helper delle credenziali

Problema: quando provi a eseguire il comando `git config` per configurare l'helper delle credenziali per comunicare con un CodeCommit repository, vedi un errore che indica che ci sono troppo pochi argomenti o un prompt d'uso che suggerisce i comandi e la sintassi di `Git config`.

Possibili correzioni: il motivo più comune di questo errore è che per il comando vengono utilizzate le virgolette singole su un sistema operativo Windows o le virgolette doppie per il comando in un sistema operativo Linux, macOS o Unix. La sintassi corretta è la seguente:

- Windows: `git config --global credential.helper '!aws codecommit credential-helper $@'`
- Linux, macOS o Unix: `git config --global credential.helper '!aws codecommit credential-helper $@'`

Vedo un errore durante la lettura del nome utente durante il tentativo di clonare un repository

Problema: quando si tenta di clonare un repository utilizzando l'helper delle credenziali, viene visualizzato un errore che indica che il sistema non è riuscito a leggere il nome utente per l'URL del repository.

Possibili correzioni: il motivo più comune di questo errore è che il file `.gitconfig` non è stato creato o configurato correttamente sul tuo computer. Apri il file `.gitconfig` e assicurati che l'helper delle credenziali sia impostato correttamente. Se utilizzi un computer che esegue Linux, macOS o Unix, assicurati anche che il valore di `$HOME` sia impostato correttamente per il tuo sistema.

Ricevo un errore di comando non trovato in Windows durante l'utilizzo dell'assistente credenziali

Problema: dopo l'aggiornamento della AWS CLI, le connessioni di supporto delle credenziali ai CodeCommit repository non funzionano. `aws codecommit credential-helper $@ get:`
`aws: command not found`

Causa: il motivo più comune di questo errore è che la tua versione AWS CLI è stata aggiornata a una versione che utilizza Python 3. Si è verificato un problema noto con il pacchetto MSI. Per verificare se disponi di una delle versioni interessate, apri una riga di comando ed esegui il comando seguente:
`aws --version`

Se la versione Python di output inizia con un 3, si dispone di una versione interessata dal problema. Per esempio:

```
aws-cli/1.16.62 Python/3.6.2 Darwin/16.7.0 botocore/1.12.52
```

Possibili correzioni: per risolvere il problema, procedi in uno dei modi seguenti:

- Installa e configura AWS CLI su Windows usando Python e pip anziché MSI. Per ulteriori informazioni, consulta [Installare Python, pip e the AWS CLI](#) su Windows.
- Modifica manualmente il file `.gitconfig` per modificare la sezione `[credential]` per puntare in modo esplicito a `aws.cmd` sul computer locale. Per esempio:

```
[credential]
  helper = !"C:\\Program Files\\Amazon\\AWSCLI\\bin\\aws.cmd\" codecommit
credential-helper $@
UseHttpPath = true
```

- Esegui il comando `git config` per aggiornare il file `.gitconfig` in modo da fare riferimento esplicitamente a `aws.cmd` e aggiorna manualmente la variabile dell'ambiente `PATH` in modo da includere il percorso al comando, secondo necessità. Per esempio:

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

Mi viene richiesto un nome utente quando mi connetto a un repository CodeCommit

Problema: quando si tenta di utilizzare l'helper delle credenziali per comunicare con un CodeCommit repository, viene visualizzato un messaggio che richiede il nome utente.

Possibili correzioni: configura il tuo AWS profilo o assicurati che il profilo che stai utilizzando sia quello configurato per l'utilizzo. CodeCommit Per ulteriori informazioni sulla configurazione, consulta [Passaggi di configurazione per le connessioni HTTPS ai AWS CodeCommit repository su Linux, macOS o Unix con l'helper delle credenziali AWS CLI](#) o [Passaggi di configurazione per le connessioni HTTPS ai AWS CodeCommit repository su Windows con l'helper delle AWS CLI credenziali](#). Per ulteriori informazioni su IAM, le chiavi di accesso e le chiavi segrete, consulta [Managing Access Keys for IAM Users and How Do I Get Credentials?](#)

Git per macOS: ho configurato correttamente l'helper delle credenziali, ma ora mi viene negato l'accesso al mio repository (403)

Problema: su macOS, l'helper delle credenziali non sembra accedere o utilizzare le credenziali come previsto. Questo può essere dovuto a due problemi diversi:

- AWS CLI È configurato per un modo Regione AWS diverso da quello in cui esiste il repository.
- Le credenziali salvate per l'utilità Accesso Portachiavi sono scadute.

Possibili correzioni: per verificare se AWS CLI è configurato per la regione corretta, esegui il `aws configure` comando ed esamina le informazioni visualizzate. Se il CodeCommit repository si trova in una posizione Regione AWS diversa da quella mostrata per la AWS CLI, è necessario eseguire il `aws configure` comando e modificare i valori in base a quelli appropriati per quella regione. Per ulteriori informazioni, consulta [Fase 1: Configurazione iniziale per CodeCommit](#).

La versione predefinita di Git rilasciata su OS X e macOS utilizza l'utilità Keychain Access per salvare le credenziali generate. Per motivi di sicurezza, la password generata per l'accesso al tuo CodeCommit repository è temporanea, quindi le credenziali memorizzate nel portachiavi smettono di funzionare dopo circa 15 minuti. Se accedi a Git solo con CodeCommit, prova quanto segue:

1. Dal terminale, eseguire il comando `git config` per individuare il file di configurazione Git (`gitconfig`) in cui è definito Accesso Portachiavi. In base al sistema locale e alle preferenze, potrebbero essere disponibili più file `gitconfig`.

```
git config -l --show-origin | grep credential
```

Nell'output di questo comando, cerca risultati simili a:

```
file://path/to/gitconfig credential.helper=osxkeychain
```

Il file elencato all'inizio di questa riga è il file di configurazione Git da modificare.

2. Per modificare il file di configurazione Git, utilizzare un editor di testo semplice o eseguire questo comando:

```
nano /usr/local/git/etc/gitconfig
```

3. Modifica la configurazione utilizzando una delle seguenti strategie:

- Commenta o elimina la sezione delle credenziali che contiene `helper = osxkeychain`. Per esempio:

```
# helper = osxkeychain
```


- Aggiorna sia la sezione di supporto alle aws credential helper osxkeychain credenziali che quella di supporto alle credenziali in modo da avere un contesto. Ad esempio, if osxkeychain viene utilizzato per autenticarsi su: GitHub

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

In questa configurazione, Git utilizzerà l'osxkeychainhelper quando l'host remoto corrisponde a "https://github.com" e l'helper per le credenziali quando l'host remoto corrisponde a "». https://git-codecommit.us-east-1.amazonaws.com

- Includi un helper di stringhe vuoto prima dell'helper per le credenziali. Ad esempio, per non utilizzare l'osxkeychainhelper quando si utilizza il profilo denominato *CodeCommitProfile* con la CLI:

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

Tip

Puoi anche configurare la riga che segue la riga che segue la riga di supporto della stringa vuota in modo che non corrisponda CodeCommit se desideri che escluda tutti i profili:

```
helper = !aws codecommit credential-helper $@
```

In alternativa, per continuare a utilizzare l'utilità Accesso Portachiavi per memorizzare nella cache le credenziali per altri repository Git, modificare l'intestazione invece di commentare la riga. Ad esempio, per consentire le credenziali memorizzate nella cache GitHub, puoi modificare l'intestazione come segue:

```
[credential "https://github.com"]
```

```
helper = osxkeychain
```

Se accedi ad altri repository con Git, puoi configurare l'utilità Keychain Access in modo che non fornisca le credenziali per i tuoi repository. CodeCommit Per configurare l'utilità Accesso Portachiavi:

1. Aprire l'utilità Accesso Portachiavi. È possibile utilizzare Finder per individuarla.
2. Cerca `git-codecommit.us-east-2.amazonaws.com` e sostituisci `us-east-2` con il luogo in Regione AWS cui si trova il repository. Evidenziare la riga, aprire il menu contestuale (pulsante destro del mouse) e scegliere Ottieni informazioni.
3. Selezionare la scheda Controllo accesso.
4. In Conferma prima di consentire l'accesso scegliere `git-credential-osxkeychain` e quindi scegliere il segno meno per rimuovere la voce dall'elenco.

Note

Dopo aver rimosso `git-credential-osxkeychain` dall'elenco, ogni volta che esegui un comando Git viene visualizzata una finestra di dialogo. Scegliere Nega per continuare. Se la finestra di dialogo popup provoca problemi, ecco alcune soluzioni alternative:

- Connettiti CodeCommit utilizzando le credenziali SSH o Git anziché l'helper delle credenziali con HTTPS. Per ulteriori informazioni, consulta [Per connessioni SSH su Linux, macOS o Unix](#) e [Configurazione per utenti HTTPS che utilizzano credenziali Git](#).
- Nell'utilità Accesso Portachiavi, nella scheda Controllo Accesso per `git-codecommit.us-east-2.amazonaws.com`, scegliere l'opzione Consenti a tutte le applicazioni di accedere a questo elemento (l'accesso a questo elemento non è limitato). Le finestre di dialogo popup non vengono visualizzate, ma le credenziali scadranno, in media, ogni 15 minuti circa e verrà quindi visualizzato un messaggio di errore 403. A quel punto, è necessario eliminare l'elemento portachiavi per ripristinare la funzionalità.
- Installare una versione di Git che non utilizza il portachiavi per impostazione predefinita.
- Prendere in considerazione l'uso di una soluzione di scripting per eliminare l'elemento del portachiavi. Per visualizzare un esempio generato dalla community di soluzione con script, consulta la sezione relativa allo [script Mac OS X per l'eliminazione periodica](#)

[delle credenziali memorizzate nella cache dello store certificati OS X](#) in [Integrazioni di prodotti e servizi](#).

Per fare in modo che Git non utilizzi affatto l'utilità Accesso Portachiavi, puoi configurare Git perché non utilizzi più `osxkeychain` come assistente credenziali. Ad esempio, se apri un terminale ed esegui il comando `git config --system credential.helper` e questo restituisce `osxkeychain` significa che Git è impostato per l'utilizzo dell'utilità Accesso Portachiavi. Puoi modificare questo comportamento eseguendo il comando:

```
git config --system --unset credential.helper
```

Tieni presente che l'esecuzione di questo comando con l'opzione `--system` modifica il comportamento di Git a livello di sistema per tutti gli utenti e ciò potrebbe avere conseguenze indesiderate per altri utenti o per altri repository se utilizzi altri servizi di repository in aggiunta a. CodeCommit. Tieni anche presente che questo approccio potrebbe richiedere l'uso di `sudo` e che l'account potrebbe non avere autorizzazioni di sistema sufficienti per applicare questa modifica. Verifica che il comando sia stato applicato correttamente eseguendo nuovamente il comando `git config --system credential.helper`. Per ulteriori informazioni, consulta [Customizing Git - Git Configuration](#) e [questo articolo in Stack Overflow](#).

Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository (403)

Problema. In Windows è possibile che l'assistente credenziali non acceda alle credenziali o che non le utilizzi come previsto. Questo può essere dovuto a problemi diversi:

- AWS CLI È configurato per un formato Regione AWS diverso da quello in cui esiste il repository.
- Per impostazione predefinita, Git per Windows installa un'utilità Git Credential Manager che non è compatibile con le CodeCommit connessioni che utilizzano l'helper delle AWS credenziali. Una volta installata, causa l'interruzione delle connessioni al repository anche se l'helper per le credenziali è stato installato con e configurato per le connessioni a. AWS CLI CodeCommit
- Alcune versioni di Git per Windows potrebbero non essere completamente conformi a [RFC 2617](#) e [RFC 4559](#), cosa che potrebbe causare problemi con le credenziali Git e l'assistente credenziali incluso in AWS CLI. Per ulteriori informazioni, consulta [Version 2.11.0\(3\) does not ask for username/password](#).

Possibili soluzioni.

- Se stai tentando di utilizzare l'helper per le credenziali incluso in AWS CLI, valuta la possibilità di connetterti con le credenziali Git tramite HTTPS invece di utilizzare l'helper delle credenziali. Le credenziali Git configurate per il tuo utente IAM sono compatibili con Git Credential Manager per Windows, a differenza dell'helper delle credenziali per. AWS CodeCommit Per ulteriori informazioni, consulta [Per utenti HTTPS che utilizzano credenziali Git](#).

Se desideri utilizzare l'helper per le credenziali, per verificare se AWS CLI è configurato correttamente Regione AWS, esegui il `aws configure` comando ed esamina le informazioni visualizzate. Se il CodeCommit repository si trova in un ambiente Regione AWS diverso da quello mostrato per la AWS CLI, è necessario eseguire il `aws configure` comando e modificare i valori in base a quelli appropriati per quella regione. Per ulteriori informazioni, consulta [Fase 1: Configurazione iniziale per CodeCommit](#).

- Se possibile, disinstallare e reinstallare Git per Windows. Quando installi Git per Windows, deseleziona la casella di controllo dell'opzione per installare l'utilità Gestione credenziali GIT. Tale strumento di gestione delle credenziali non è compatibile con l'assistente credenziali per AWS CodeCommit. Se si è eseguita l'installazione di Gestione credenziali GIT o di un'altra utilità di gestione delle credenziali e non si desidera effettuarne la disinstallazione, è possibile modificare il file `.gitconfig` e aggiungere la gestione delle credenziali specifiche per CodeCommit:
 1. Apri il Pannello di controllo, scegli Gestione credenziali e rimuovi le credenziali archiviate per. CodeCommit
 2. Aprire il file `.gitconfig` in qualsiasi editor di testo semplice, ad esempio Blocco note.

Note

Se si utilizzano più profili Git, potrebbero essere disponibili file `.gitconfig` globali e locali. Assicurarsi di modificare il file appropriato.

3. Aggiungere la sezione seguente al file `.gitconfig`:

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

4. Salvare il file e quindi aprire una nuova sessione della riga di comando prima di tentare di connettersi di nuovo.

Puoi utilizzare questo approccio anche se desideri utilizzare l'helper per le credenziali per AWS CodeCommit quando ti connetti ai CodeCommit repository e un altro sistema di gestione delle credenziali quando ti connetti ad altri repository ospitati, come i repository. GitHub

Per reimpostare l'assistente credenziali utilizzato per impostazione predefinita, puoi usare l'opzione `--system` anziché `--global` o `--local` quando esegui il comando `git config`.

- Se si utilizzano le credenziali Git su un computer Windows, è possibile provare a risolvere eventuali problemi di non conformità a RFC includendo il nome utente delle credenziali Git come parte della stringa di connessione. Ad esempio, per risolvere il problema e clonare un repository denominato *MyDemoRepo* nella regione degli Stati Uniti orientali (Ohio):

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

Note

Questo approccio non funziona se il nome utente nelle credenziali Git contiene un carattere `@`. È necessario applicare la codifica URL (operazione nota anche come escape di URL o [codifica percentuale](#)) al carattere.

Risoluzione dei problemi relativi ai client Git e AWS CodeCommit

Le informazioni seguenti possono essere utili per risolvere problemi comuni durante l'uso di Git con i repository AWS CodeCommit. Per la risoluzione di problemi correlate ai client Git durante l'utilizzo di HTTPS o SSH, consulta anche [Risoluzione dei problemi relativi alle credenziali Git \(HTTPS\)](#), [Risoluzione dei problemi relativi alle connessioni SSH](#) e [Risoluzione dei problemi relativi all'helper delle credenziali \(HTTPS\)](#).

Argomenti

- [Errore Git: Errore: RPC fallito; risultato = 56, codice HTTP = 200 fatale: l'estremità remota si è bloccata in modo imprevisto](#)
- [Errore Git: troppi comandi di aggiornamento riferimenti](#)
- [Errore Git: il push tramite HTTPS non funziona in alcune versioni di Git](#)
- [Errore Git: "gnutls_handshake\(\) failed" \("gnutls_handshake\(\) non riuscito"\)](#)

- [Errore Git: Git non riesce a trovare il CodeCommit repository o non ha il permesso di accedere al repository](#)
- [Git in Windows: nessun metodo di autenticazione supportato disponibile \(publickey\)](#)

Errore Git: Errore: RPC fallito; risultato = 56, codice HTTP = 200 fatale: l'estremità remota si è bloccata in modo imprevisto

Problema. Quando si esegue il push di una modifica di grandi dimensioni, di un numero elevato di modifiche o di un repository di grandi dimensioni, le connessioni HTTPS di lunga durata vengono spesso terminate prematuramente a causa di problemi di rete o delle impostazioni del firewall.

Possibili soluzioni. Esegui invece il push con SSH, oppure in caso di migrazione di un repository di grandi dimensioni, segui i passaggi descritti in [Esegui la migrazione di un repository in modo incrementale](#). Assicurati inoltre di non superare i limiti di dimensioni stabiliti per i singoli file. Per ulteriori informazioni, consulta [Quote](#).

Errore Git: troppi comandi di aggiornamento riferimenti

Problema. Il numero massimo di aggiornamenti dei riferimenti per push è 4.000. Questo errore viene visualizzato quando il push contiene più di 4.000 aggiornamenti dei riferimenti.

Possibili soluzioni. Prova a eseguire il push di rami e tag singolarmente con `git push --all` e `git push --tags`. In presenza di troppi tag, suddividi i tag in più push. Per ulteriori informazioni, consulta [Quote](#).

Errore Git: il push tramite HTTPS non funziona in alcune versioni di Git

Problema. Un problema dell'aggiornamento curl alla versione 7.41.0 impedisce il completamento dell'autenticazione digest basata su SSPI. Le versioni di Git interessate di cui si ha conoscenza includono 1.9.5.msysgit.1. Alcune versioni di Git per Windows potrebbero non essere completamente conformi a [RFC 2617](#) e [RFC 4559](#), con potenziali problemi nelle connessioni HTTPS in cui vengono usate credenziali Git o l'assistente credenziali incluso in AWS CLI.

Possibili soluzioni. Controlla i problemi noti della versione di Git in uso oppure utilizza una versione precedente o successiva. [Per ulteriori informazioni su msysgit, vedi Push to HTTPS Is Broken nei forum.](#) GitHub Per ulteriori informazioni sui problemi della versione di Git per Windows, consulta [Version 2.11.0\(3\) does not ask for username/password.](#)

Errore Git: "gnutls_handshake() failed" ("gnutls_handshake() non riuscito")

Problema: in Linux, quando si tenta di utilizzare Git per comunicare con un CodeCommit repository, viene visualizzato un messaggio di errore contenente la frase `error: gnutls_handshake() failed`.

Possibili soluzioni. Compila Git su OpenSSL. Per un approccio, consulta ["Error: gnutls_handshake\(\) failed" When Connecting to HTTPS Servers](#) nei forum Ask Ubuntu.

In alternativa, usa SSH anziché HTTPS per comunicare con CodeCommit i repository.

Errore Git: Git non riesce a trovare il CodeCommit repository o non ha il permesso di accedere al repository

Problema. I tentativi di connessione possono non riuscire a causa di una barra finale nella stringa di connessione.

Possibili soluzioni: Verifica di avere specificato il nome e la stringa di connessione corretti per il repository e che non siano presenti barre finali. Per ulteriori informazioni, consulta [Connessione a un repository](#).

Git in Windows: nessun metodo di autenticazione supportato disponibile (publickey)

Problema. Dopo che è stato configurato l'accesso SSH per Windows viene visualizzato un errore di accesso negato quando si tenta di usare comandi quali `git pull`, `git push` o `git clone`.

Possibili soluzioni. La causa più comune di questo errore è la presenza nel computer di una variabile di ambiente `GIT_SSH` configurata per supportare un'altra utilità di connessione, ad esempio PuTTY. Per risolvere questo problema, prova una delle soluzioni seguenti:

- Aprire un emulatore Bash e aggiungere il parametro `GIT_SSH_COMMAND="ssh"` prima del comando Git. Ad esempio, se si sta tentando di clonare un repository, anziché eseguire `git clone` `ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo`, eseguire:

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

- Rinominare o eliminare la variabile di ambiente GIT_SSH se non la si utilizza più. Quindi aprire un nuovo prompt dei comandi o una nuova sessione dell'emulatore Bash e riprovare a eseguire il comando.

Per ulteriori informazioni sulla risoluzione dei problemi di Git in Windows quando si utilizza SSH, consulta [Risoluzione dei problemi relativi alle connessioni SSH](#).

Risoluzione degli errori di accesso e AWS CodeCommit

Le informazioni seguenti possono essere utili per risolvere gli errori di accesso durante la connessione ai repository AWS CodeCommit .

Argomenti

- [Errore di accesso: mi vengono richiesti un nome utente e una password quando mi connetto a un CodeCommit repository da Windows](#)
- [Errore di accesso: chiave pubblica rifiutata durante la connessione a un repository CodeCommit](#)
- [Errore di accesso: messaggio «Rate exceeded» o «429" durante la connessione a un repository CodeCommit](#)

Errore di accesso: mi vengono richiesti un nome utente e una password quando mi connetto a un CodeCommit repository da Windows

Problema: quando si tenta di utilizzare Git per comunicare con un CodeCommit repository, viene visualizzata una finestra di dialogo che richiede il nome utente e la password.

Possibili soluzioni: potrebbe trattarsi del sistema di gestione delle credenziali integrato per Windows. In base alla configurazione, esegui una delle seguenti operazioni:

- Se stai utilizzando HTTPS con le credenziali Git, queste non sono ancora memorizzate nel sistema. Fornisci le credenziali Git e continua. La richiesta non dovrebbe venire più visualizzata. Per ulteriori informazioni, consulta [Per utenti HTTPS che utilizzano credenziali Git](#).
- Se utilizzi HTTPS con l'helper delle credenziali per AWS CodeCommit, questo non è compatibile con il sistema di gestione delle credenziali di Windows. Seleziona Annulla.

Ciò potrebbe anche indicare che al momento dell'installazione di Git per Windows hai installato Git Credential Manager. Git Credential Manager non è compatibile con l'helper per le credenziali

CodeCommit incluso in. AWS CLI Considera la disinstallazione di Git Credential Manager. È inoltre possibile installare e configurare git-remote-codecommit in alternativa all'utilizzo dell'assistente credenziali per CodeCommit.

Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit con git-remote-codecommit](#), [Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI](#) e [Git per Windows: ho installato Git per Windows, ma mi viene negato l'accesso al mio repository \(403\)](#).

Errore di accesso: chiave pubblica rifiutata durante la connessione a un repository CodeCommit

Problema: quando si tenta di utilizzare un endpoint SSH per comunicare con un CodeCommit repository, viene visualizzato un messaggio di errore contenente la frase. `Error: public key denied`

Possibili soluzioni: il motivo più comune di questo errore è il mancato completamento della configurazione delle connessioni SSH. Configura una coppia di chiavi SSH pubblica e privata, quindi associa la chiave pubblica al tuo utente IAM. Per ulteriori informazioni sulla configurazione di SSH, consulta [Per connessioni SSH su Linux, macOS o Unix](#) e [Per le connessioni SSH su Windows](#).

Errore di accesso: messaggio «Rate exceeded» o «429» durante la connessione a un repository CodeCommit

Problema. Quando provi a comunicare con un repository CodeCommit, viene visualizzato un messaggio che riporta "Rate Exceeded (Velocità superata)" o il codice errore "429". La comunicazione rallenta significativamente o non riesce.

Causa: tutte le chiamate CodeCommit, provenienti da un'applicazione, da un client Git o dal Console di gestione AWS, sono soggette a un numero massimo di richieste al secondo e a un totale di richieste attive. AWS CLI Non puoi superare la frequenza massima di richieste consentita per un account Amazon Web Services in nessun caso Regione AWS. Se le richieste superano la velocità massima, ricevi un errore e altre chiamate vengono temporaneamente limitate per il tuo account Amazon Web Services. Durante il periodo di limitazione delle richieste, le connessioni ad CodeCommit sono rallentate e potrebbero non riuscire.

Possibili correzioni: adotta misure per ridurre il numero di connessioni o chiamate CodeCommit o per distribuire le richieste. Alcuni approcci da prendere in considerazione:

- Implementare jitter nelle richieste, in particolare nelle richieste di polling periodiche

Se disponi di un'applicazione che effettua il polling CodeCommit periodico e questa applicazione è in esecuzione su più EC2 istanze Amazon, inserisci il jitter (una quantità di ritardo casuale) in modo che EC2 istanze Amazon diverse non eseguano il polling nello stesso secondo. È consigliabile un numero casuale da 0 a 59 secondi per distribuire meccanismi di polling in maniera uniforme su un intervallo temporale di un minuto.

- Utilizzare un'architettura basata sugli eventi piuttosto che il polling

Anziché il polling, utilizza un'architettura basata sugli eventi, in modo che le chiamate vengano effettuate solo quando si verifica un evento. Prendi in considerazione l'utilizzo delle notifiche CloudWatch Events for [AWS CodeCommit Events per attivare il tuo flusso di lavoro](#).

- Implementa tentativi di errore e backoff esponenziali per azioni Git automatizzate APIs

Ripetizione dei tentativi in caso di errore e backoff esponenziali possono consentire di limitare la frequenza delle chiamate. Ogni SDK AWS implementa automaticamente la logica di ripetizione dei tentativi e gli algoritmi di backoff esponenziale. Per Git push e Git pull automatizzati, potrebbe essere necessario implementare la propria logica di ripetizione dei tentativi. Per ulteriori informazioni, consulta [Error Retries](#) and Exponential Backoff in AWS.

- Richiedi un aumento della quota di CodeCommit servizio nel AWS Support Center

Per ottenere un aumento delle restrizioni dei servizi, devi confermare che hai già seguito i suggerimenti utili offerti qui, tra cui l'implementazione della ripetizione dei tentativi in caso di errore o dei metodi di backoff esponenziale. Nella richiesta, devi anche fornire l' Regione AWS account Amazon Web Services e il periodo di tempo interessati dai problemi di limitazione.

Risoluzione degli errori di configurazione e AWS CodeCommit

Le seguenti informazioni possono aiutarti a risolvere gli errori di configurazione che potresti riscontrare durante la connessione ai AWS CodeCommit repository.

Argomenti

- [Errore di configurazione: impossibile configurare AWS CLI le credenziali su macOS](#)

Errore di configurazione: impossibile configurare AWS CLI le credenziali su macOS

Problema: quando si esegue `aws configure` la configurazione AWS CLI, viene visualizzato un `ConfigParseError` messaggio.

Possibili soluzioni: la causa più comune di questo errore è il fatto che esiste già un file delle credenziali. Sfoglia fino a `~/.aws` e cerca il file `credentials`. Assegna un nuovo nome o elimina il file, quindi esegui nuovamente `aws configure`.

Risoluzione degli errori della console e AWS CodeCommit

Le informazioni seguenti possono essere utili per risolvere gli errori della console durante l'uso dei repository AWS CodeCommit .

Argomenti

- [Errore di accesso: accesso negato alla chiave di crittografia per un CodeCommit repository dalla console o AWS CLI](#)
- [Errore di crittografia: il repository non può essere decrittografato](#)
- [Errore della console: impossibile sfogliare il codice in un repository dalla console CodeCommit](#)
- [Errore di visualizzazione: impossibile visualizzare un file o un confronto tra file](#)

Errore di accesso: accesso negato alla chiave di crittografia per un CodeCommit repository dalla console o AWS CLI

Problema: quando si tenta di accedere CodeCommit dalla console o dal AWS CLI, viene visualizzato un messaggio di errore contenente la frase `EncryptionKeyAccessDeniedException` o `User is not authorized for the KMS default key for CodeCommit 'aws/codecommit' in your account`.

Possibili correzioni: La causa più comune di questo errore è che il tuo account Amazon Web Services non è abbonato AWS Key Management Service, il che è necessario per. CodeCommit Apri la AWS KMS console, scegli le chiavi AWS gestite, quindi scegli Inizia subito. Se vedi un messaggio che indica che non sei attualmente abbonato al AWS Key Management Service servizio, segui le istruzioni in quella pagina per iscriverti. Per ulteriori informazioni su CodeCommit e AWS Key Management Service, consulta [AWS KMS e crittografia](#).

Errore di crittografia: il repository non può essere decrittografato

Problema: quando si tenta di accedere a un CodeCommit repository dalla console o dal AWS CLI, viene visualizzato un messaggio di errore contenente la frase. Repository can't be decrypted

Possibili correzioni: La causa più comune di questo errore è che la AWS KMS chiave utilizzata per crittografare e decrittografare i dati per questo repository non è attiva o è in attesa di eliminazione. È necessaria una chiave di accesso attiva Chiave gestita da AWS o gestita dal cliente per. AWS Key Management Service CodeCommit Apri la AWS KMS console, scegli Chiavi gestite da AWSLe chiavi gestite dal cliente e assicurati che la chiave utilizzata per il repository sia presente nel luogo in Regione AWS cui si trova il repository e che il suo stato sia Attivo. Per ulteriori informazioni su CodeCommit and AWS Key Management Service, consulta. [AWS KMS e crittografia](#)

Important

Se la chiave utilizzata per crittografare e decrittografare i dati del repository è stata eliminata definitivamente o è altrimenti inaccessibile, non è possibile accedere ai dati nei repository crittografati con quella chiave.

Errore della console: impossibile sfogliare il codice in un repository dalla console CodeCommit

Problema. Quando si tenta di esplorare il contenuto di un repository dalla console viene visualizzato un messaggio di errore che nega l'accesso.

Possibili correzioni: La causa più comune di questo errore è che una policy IAM applicata al tuo account Amazon Web Services nega una o più delle autorizzazioni richieste per la navigazione del codice dalla console. CodeCommit Per ulteriori informazioni sulle autorizzazioni di CodeCommit accesso e sulla navigazione, consulta. [Autenticazione e controllo degli accessi per AWS CodeCommit](#)

Errore di visualizzazione: impossibile visualizzare un file o un confronto tra file

Problema: quando si tenta di visualizzare un file o un confronto tra due versioni di un file nella CodeCommit console, viene visualizzato un errore che indica che il file o la differenza sono troppo grandi per essere visualizzati.

Possibili correzioni: la causa più comune di questo errore è che il file è troppo grande per essere visualizzato, contiene una o più righe che superano il limite di caratteri per una singola riga in un file o che la differenza tra le due versioni del file supera il limite di riga. Per ulteriori informazioni, consulta [Quote](#). Per visualizzare il file o le differenze tra le versioni del file, puoi aprire il file localmente nel tuo IDE preferito, usare uno strumento Git diff o eseguire il git diff comando.

Risoluzione dei problemi relativi ai trigger e AWS CodeCommit

Le informazioni seguenti possono risultare utili per risolvere i problemi relativi ai trigger in AWS CodeCommit.

Argomenti

- [Errore del trigger: il trigger di un repository non viene eseguito quando previsto](#)

Errore del trigger: il trigger di un repository non viene eseguito quando previsto

Problema. Uno o più trigger configurati per un repository non sembrano essere eseguiti o non vengono eseguiti come previsto.

Possibili soluzioni. Se la destinazione del trigger è una funzione AWS Lambda , verifica di aver configurato la policy delle risorse della funzione per l'accesso da parte di CodeCommit.

In alternativa, modifica il trigger e verifica che gli eventi per i quali vuoi attivare le azioni siano stati selezionati e che i rami del trigger includano il ramo in cui vuoi visualizzare le risposte alle azioni. Prova a impostare il trigger su All repository events (Tutti gli eventi del repository) e All branches (Tutti i rami) e quindi testa il trigger. Per ulteriori informazioni, consulta [Modifica i trigger per un repository](#).

Attiva il debug

Problema. Vorrei attivare il debug per ottenere una maggiore quantità di informazioni sul mio repository e su come Git esegue i comandi.

Possibili soluzioni. Prova questa procedura:

1. Al terminale o al prompt dei comandi eseguire i comandi seguenti sul computer locale prima dei comandi Git:

Su Linux, macOS o Unix:

```
export GIT_TRACE_PACKET=1
export GIT_TRACE=1
export GIT_CURL_VERBOSE=1
```

In Windows:

```
set GIT_TRACE_PACKET=1
set GIT_TRACE=1
set GIT_CURL_VERBOSE=1
```

Note

L'impostazione di `GIT_CURL_VERBOSE` è utile solo per le connessioni HTTPS. SSH non utilizza la libreria `libcurl`.

2. Per ottenere maggiori informazioni sul tuo repository Git, ti consigliamo di installare l'ultima versione di [git-sizer](#). Segui le istruzioni per installare l'utilità appropriata al tuo sistema operativo e al tuo ambiente. Una volta installata, dalla riga di comando o dal terminale, modificate le directory nel vostro repository locale, quindi eseguite il comando seguente:

```
git-sizer --verbose
```

 Tip

Valuta la possibilità di salvare l'output del comando in un file in modo da poterlo condividere facilmente con altri utenti durante la risoluzione dei problemi, in particolare nel tempo.

AWS CodeCommit riferimento

I seguenti argomenti di riferimento possono aiutarti a comprendere meglio CodeCommit, Git Regioni AWS, i limiti del servizio e altro ancora.

Argomenti

- [Regioni ed endpoint di connessione Git per AWS CodeCommit](#)
- [Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia](#)
- [Quote in AWS CodeCommit](#)
- [AWS CodeCommit riferimento alla riga di comando](#)
- [Comandi Git di base](#)

Regioni ed endpoint di connessione Git per AWS CodeCommit

Ogni CodeCommit repository è associato a un. Regione AWS CodeCommit offre endpoint regionali per inoltrare le richieste al servizio. Inoltre, CodeCommit fornisce endpoint di connessione Git per i protocolli SSH e HTTPS in ogni regione in cui CodeCommit è disponibile.

Tutti gli esempi di questa guida utilizzano lo stesso URL di endpoint per Git negli Stati Uniti orientali (Ohio): `git-codecommit.us-east-2.amazonaws.com` Tuttavia, quando usi Git e configuri le tue connessioni, assicurati di scegliere l'endpoint di connessione Git che corrisponda a quello Regione AWS che ospita il tuo CodeCommit repository. Ad esempio, se desideri stabilire una connessione a un repository negli Stati Uniti orientali (Virginia settentrionale), utilizza l'URL dell'endpoint di `git-codecommit.us-east-1.amazonaws.com` Questo vale anche per le chiamate API. Quando effettui connessioni a un CodeCommit repository con AWS CLI o il SDKs, assicurati di utilizzare l'endpoint regionale corretto per il repository.

Argomenti

- [Supportato per Regioni AWS CodeCommit](#)
- [Endpoint di connessione Git](#)
- [Impronte digitali del server per CodeCommit](#)

Supportato per Regioni AWS CodeCommit

È possibile creare e utilizzare i CodeCommit repository nei seguenti modi: Regioni AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Parigi)
- Europa (Francoforte)
- Europa (Stoccolma)
- Europa (Milano)
- Africa (Città del Capo)
- Israele (Tel Aviv)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Giacarta)
- Medio Oriente (Emirati Arabi Uniti)
- Asia Pacifico (Seoul)
- Asia Pacifico (Osaka)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Hyderabad)
- Asia Pacifico (Hong Kong)
- Sud America (San Paolo)
- Medio Oriente (Bahrein)
- Canada (Centrale)
- Cina (Pechino)
- Cina (Ningxia)
- AWS GovCloud (Stati Uniti occidentali)
- AWS GovCloud (Stati Uniti orientali)

CodeCommit ha aggiunto il supporto per lo standard governativo Federal Information Processing Standard (FIPS) Publication 140-2 in alcune regioni. Per ulteriori informazioni su FIPS e gli endpoint FIPS, consulta la [panoramica sullo standard Federal Information Processing Standard \(FIPS\) 140-2](#). Per gli endpoint di connessione Git che supportano FIPS, consulta [Endpoint di connessione Git](#).

Per ulteriori informazioni sugli endpoint regionali AWS CLI, sui servizi e sulle chiamate API verso CodeCommit, consulta [AWS CodeCommit Endpoints](#) and Quotas.

Endpoint di connessione Git

Usa quanto segue URLs quando configuri le connessioni Git ai CodeCommit repository:

Endpoint di connessione Git per AWS CodeCommit

Nome Regione	Regione	URL dell'endpoint	Protocollo
Stati Uniti orientali (Ohio)	us-east-2	https://git-codecommit.us-east-2.amazonaws.com	HTTPS
Stati Uniti orientali (Ohio)	us-east-2	ssh://git-codecommit.us-east-2.amazonaws.com	SSH
Stati Uniti orientali (Ohio)	us-east-2	https://git-codecommit-fips.us-east-2.amazonaws.com	HTTPS
Stati Uniti orientali (Virginia settentrionale)	us-east-1	https://git-codecommit.us-east-1.amazonaws.com	HTTPS
Stati Uniti orientali (Virginia settentrionale)	us-east-1	ssh://git-codecommit.us-east-1.amazonaws.com	SSH
Stati Uniti orientali (Virginia settentrionale)	us-east-1	https://git-codecommit-fips.us-east-1.amazonaws.com	HTTPS

Nome Regione	Regione	URL dell'endpoint	Protocollo
Stati Uniti occidentali (Oregon)	us-west-2	https://git-codecommit.us-west-2.amazonaws.com	HTTPS
Stati Uniti occidentali (Oregon)	us-west-2	ssh://git-codecommit.us-west-2.amazonaws.com	SSH
Stati Uniti occidentali (Oregon)	us-west-2	https://git-codecommit-fips.us-west-2.amazonaws.com	HTTPS
Stati Uniti occidentali (California settentrionale)	us-west-1	https://git-codecommit.us-west-1.amazonaws.com	HTTPS
Stati Uniti occidentali (California settentrionale)	us-west-1	ssh://git-codecommit.us-west-1.amazonaws.com	SSH
Stati Uniti occidentali (California settentrionale)	us-west-1	https://git-codecommit-fips.us-west-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	https://git-codecommit.eu-west-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	ssh://git-codecommit.eu-west-1.amazonaws.com	SSH
Asia Pacifico (Tokyo)	ap-northeast-1	https://git-codecommit.ap-northeast-1.amazonaws.com	HTTPS

Nome Regione	Regione	URL dell'endpoint	Protocollo
Asia Pacifico (Tokyo)	ap-northeast-1	ssh://git-codecommit.ap-northeast-1.amazonaws.com	SSH
Asia Pacifico (Singapore)	ap-southeast-1	https://git-codecommit.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacifico (Singapore)	ap-southeast-1	ssh://git-codecommit.ap-southeast-1.amazonaws.com	SSH
Asia Pacifico (Sydney)	ap-southeast-2	https://git-codecommit.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacifico (Sydney)	ap-southeast-2	ssh://git-codecommit.ap-southeast-2.amazonaws.com	SSH
Asia Pacifico (Giacarta)	ap-southeast-3	https://git-codecommit.ap-southeast-3.amazonaws.com	HTTPS
Asia Pacifico (Giacarta)	ap-southeast-3	ssh://git-codecommit.ap-southeast-3.amazonaws.com	SSH
Medio Oriente (Emirati Arabi Uniti)	me-central-1	https://git-codecommit.me-central-1.amazonaws.com	HTTPS
Medio Oriente (Emirati Arabi Uniti)	me-central-1	ssh://git-codecommit.me-central-1.amazonaws.com	SSH

Nome Regione	Regione	URL dell'endpoint	Protocollo
Europa (Francoforte)	eu-central-1	https://git-codecommit.eu-central-1.amazonaws.com	HTTPS
Europa (Francoforte)	eu-central-1	ssh://git-codecommit.eu-central-1.amazonaws.com	SSH
Asia Pacific (Seoul)	ap-northeast-2	https://git-codecommit.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ssh://git-codecommit.ap-northeast-2.amazonaws.com	SSH
Sud America (San Paolo)	sa-east-1	https://git-codecommit.sa-east-1.amazonaws.com	HTTPS
Sud America (San Paolo)	sa-east-1	ssh://git-codecommit.sa-east-1.amazonaws.com	SSH
Europe (London)	eu-west-2	https://git-codecommit.eu-west-2.amazonaws.com	HTTPS
Europe (London)	eu-west-2	ssh://git-codecommit.eu-west-2.amazonaws.com	SSH
Asia Pacifico (Mumbai)	ap-south-1	https://git-codecommit.ap-south-1.amazonaws.com	HTTPS

Nome Regione	Regione	URL dell'endpoint	Protocollo
Asia Pacifico (Mumbai)	ap-south-1	ssh://git-codecommit.ap-south-1.amazonaws.com	SSH
Asia Pacifico (Hyderabad)	ap-south-2	https://git-codecommit.ap-south-2.amazonaws.com	HTTPS
Asia Pacifico (Hyderabad)	ap-south-2	ssh://git-codecommit.ap-south-2.amazonaws.com	SSH
Canada (Centrale)	ca-central-1	https://git-codecommit.ca-central-1.amazonaws.com	HTTPS
Canada (Centrale)	ca-central-1	ssh://git-codecommit.ca-central-1.amazonaws.com	SSH
Canada (Centrale)	ca-central-1	https://git-codecommit-fips.ca-central-1.amazonaws.com	HTTPS
Europa (Parigi)	eu-west-3	https://git-codecommit.eu-west-3.amazonaws.com	HTTPS
Europa (Parigi)	eu-west-3	ssh://git-codecommit.eu-west-3.amazonaws.com	SSH
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	https://git-codecommit-northeast-1.amazonaws.com	HTTPS

Nome Regione	Regione	URL dell'endpoint	Protocollo
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	ssh: //git-codecommit. us-gov-west-1. amazonaws.com	SSH
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	https://git-codecommit- fips. us-gov-west-1. amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	https://git-codeco mnebbia. us-gov-ea st-1. amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	ssh: //git-codecommit. us-gov-east-1. amazonaws.com	SSH
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	https://git-codecommit- fips. us-gov-east-1. amazonaws.com	HTTPS
Europa (Stoccolma)	eu-north-1	https://git-codeco mmit.eu-north-1.am azonaws.com	HTTPS
Europa (Stoccolma)	eu-north-1	ssh://git-codecomm it.eu-north-1.amaz onaws.com	SSH
Medio Oriente (Bahrein)	me-south-1	https://git-codeco mmit.me-south-1.am azonaws.com	HTTPS
Medio Oriente (Bahrein)	me-south-1	ssh://git-codecomm it.me-south-1.amaz onaws.com	SSH

Nome Regione	Regione	URL dell'endpoint	Protocollo
Asia Pacific (Hong Kong)	ap-east-1	https://git-codecommit.ap-east-1.amazonaws.com	HTTPS
Asia Pacific (Hong Kong)	ap-east-1	ssh://git-codecommit.ap-east-1.amazonaws.com	SSH
Cina (Pechino)	cn-north-1	https://git-codecommit.cn-north-1.amazonaws.com.cn	HTTPS
Cina (Pechino)	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
Europe (Milan)	eu-south-1	https://git-codecommit.eu-south-1.amazonaws.com	HTTPS
Europe (Milan)	eu-south-1	ssh://git-codecommit.eu-south-1.amazonaws.com	SSH
Asia Pacifico (Osaka-Locale)	ap-northeast-3	https://git-codecommit.ap-northeast-3.amazonaws.com	HTTPS

Nome Regione	Regione	URL dell'endpoint	Protocollo
Asia Pacifico (Osaka-Locale)	ap-northeast-3	ssh://git-codecommit.ap-northeast-3.amazonaws.com	SSH
Africa (Città del Capo)	af-south-1	https://git-codecommit.af-south-1.amazonaws.com	HTTPS
Africa (Città del Capo)	af-south-1	ssh://git-codecommit.af-south-1.amazonaws.com	SSH
Israele (Tel Aviv)	il-central-1	https://git-codecommit.il-central-1.amazonaws.com	HTTPS
Israele (Tel Aviv)	il-central-1	ssh://git-codecommit.il-central-1.amazonaws.com	SSH

Impronte digitali del server per CodeCommit

La tabella seguente elenca le impronte pubbliche per gli endpoint di connessione Git in CodeCommit. Queste impronte del server vengono visualizzate nell'ambito del processo di verifica per aggiungere un endpoint ai file hosts noti.

Impronte digitali pubbliche per CodeCommit

Server	Tipo di hash crittografico	Impronta digitale
git-codecommit.us-east-2.amazonaws.com	MD5	a9:6d:03:ed:08:42: 21:be:06:e1:e0:2a: d1:75:31:5e

Server	Tipo di hash crittografico	Impronta digitale
git-codecommit.us-east-2.amazonaws.com	SHA256	31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UEs56fG6ZIZQ
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+8isPtotBoY
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	0pJx9SQpkbPUAHwy58UVIq0IHcyo1fwCp00uVgcAWPo
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	tKjRk0L8dmJyTmSbeSdN1S8F/f0iq13RlvqgTOP1UyQ
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/bnBybzhiuu4dWpBJtXPf7E30jHU7se40w

Server	Tipo di hash crittografico	Impronta digitale
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 Qr1lTZA
git-codecommit.eu-central-1.amazonaws.com	MD5	74:5a:e8:02:fc:b2: 9c:06:10:b4:78:84: 65:94:22:2d

Server	Tipo di hash crittografico	Impronta digitale
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 K0mxtfXBHzAZd8Eya 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

Server	Tipo di hash crittografico	Impronta digitale
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
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

Server	Tipo di hash crittografico	Impronta digitale
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+V8jl7iuA 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
git-codecommit.ap-east-1.amazonaws.com	SHA256	LafadYwUYW8h0NoTRp objbjNs9IRnbEwHtezD 3aAIBX0

Server	Tipo di hash crittografico	Impronta digitale
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 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

Server	Tipo di hash crittografico	Impronta digitale
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
git-codecommit.il-central-1.amazonaws.com	SHA256	uFxhp51kUWh1eTLeYb xQVYm4RnNLNZ5Dbdm1 cgdS1/8

Utilizzo AWS CodeCommit con gli endpoint VPC dell'interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e CodeCommit. È possibile utilizzare questa connessione per abilitare CodeCommit per comunicare con le risorse nel VPC senza accedere all'Internet pubblico.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Con gli endpoint VPC, il routing tra il VPC e i AWS servizi viene gestito dalla AWS rete e puoi utilizzare le policy IAM per controllare l'accesso alle risorse del servizio.

Per connettere il tuo VPC CodeCommit, definisci un'interfaccia VPC endpoint per CodeCommit. Un endpoint di interfaccia è un'interfaccia di rete elastica con un indirizzo IP privato che funge da punto di ingresso per il traffico destinato a un servizio supportato AWS. L'endpoint fornisce una connettività affidabile e scalabile CodeCommit senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta [Che cos'è Amazon VPC?](#) nella Guida per l'utente Amazon VPC.

Note

Altri AWS servizi che forniscono supporto VPC e si integrano con CodeCommit, ad esempio AWS CodePipeline, potrebbero non supportare l'utilizzo degli endpoint Amazon VPC per tale integrazione. Ad esempio, il traffico tra CodePipeline e CodeCommit non può essere limitato

all'intervallo della sottorete VPC. I servizi che supportano l'integrazione, ad esempio [AWS Cloud9](#), potrebbero richiedere servizi aggiuntivi come AWS Systems Manager

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta [AWS PrivateLink](#).

Le fasi seguenti sono per gli utenti Amazon VPC. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base](#) nella Guida per l'utente di Amazon VPC.

Disponibilità

CodeCommit attualmente supporta gli endpoint VPC nei seguenti casi: Regioni AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Parigi)
- Europa (Francoforte)
- Europa (Stoccolma)
- Europa (Milano)
- Africa (Città del Capo)
- Israele (Tel Aviv)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Giacarta)
- Medio Oriente (Emirati Arabi Uniti)
- Asia Pacifico (Seoul)

- Asia Pacifico (Osaka)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Hyderabad)
- Asia Pacifico (Hong Kong)
- Sud America (San Paolo)
- Medio Oriente (Bahrein)
- Canada (Centrale)
- Cina (Pechino)
- Cina (Ningxia)
- AWS GovCloud (Stati Uniti occidentali)
- AWS GovCloud (Stati Uniti orientali)

Crea endpoint VPC per CodeCommit

Per iniziare a utilizzarlo CodeCommit con il tuo VPC, crea un endpoint VPC di interfaccia per. CodeCommit CodeCommit richiede endpoint separati per le operazioni Git e per le operazioni CodeCommit API. A seconda delle esigenze aziendali, potrebbe essere necessario creare più endpoint VPC. Quando crei un endpoint VPC per CodeCommit, scegli AWS Servizi e in Nome servizio scegli una delle seguenti opzioni:

- `com.amazonaws. region.git-codecommit`: scegli questa opzione se desideri creare un endpoint VPC per le operazioni Git con i repository. CodeCommit Ad esempio, scegli questa opzione se gli utenti utilizzano un client Git e comandi come `git pullgit commit`, e `git push` quando interagiscono con i CodeCommit repository.
- `com.amazonaws. region.git-codecommit-fips`: Scegli questa opzione se desideri creare un endpoint VPC per le operazioni Git con CodeCommit repository conformi allo standard governativo statunitense Federal Information Processing Standard (FIPS) Publication 140-2.

Note

Gli endpoint FIPS per Git non sono disponibili in tutte le AWS regioni. Per ulteriori informazioni, consulta [Endpoint di connessione Git](#).

- `com.amazonaws. region.codecommit`: scegli questa opzione se desideri creare un endpoint VPC per le operazioni API. CodeCommit Ad esempio, scegli questa opzione se gli utenti utilizzano

l' AWS CLI, l' CodeCommit API o il con cui interagire AWS SDKs CodeCommit per operazioni come `CreateRepository`, e. `ListRepositories` `PutFile`

- `com.amazonaws. region.codecommit-fips`: scegli questa opzione se desideri creare un endpoint VPC per le operazioni CodeCommit API conforme allo standard governativo statunitense Federal Information Processing Standard (FIPS) Publication 140-2.

Note

Gli endpoint FIPS non sono disponibili in tutte le regioni. AWS Per ulteriori informazioni, vedere la voce relativa alla panoramica del AWS CodeCommit [Federal Information Processing Standard \(FIPS\) 140-2](#).

Creazione di una policy di endpoint VPC per CodeCommit.

Puoi creare una policy per gli endpoint Amazon VPC per la CodeCommit quale puoi specificare:

- Il principale che può eseguire azioni.
- Le operazioni che possono essere eseguite.
- Le risorse su cui è possibile eseguire le operazioni.

Ad esempio, un'azienda può limitare l'accesso ai repository all'intervallo di indirizzi di rete per un VPC. Un esempio di questo tipo di policy è disponibile qui: [Esempio 3: consentire a un utente che si connette da un intervallo di indirizzi IP specificato di accedere a un repository](#). L'azienda ha configurato due endpoint Git VPC per la regione Stati Uniti orientali (Ohio): e. `com.amazonaws.us-east-2.codecommit` `com.amazonaws.us-east-2.git-codecommit-fips` Vogliono consentire l'invio di codice a un CodeCommit repository denominato *MyDemoRepo* solo sull'endpoint conforme allo standard FIPS. Per applicare questa regola, occorre configurare una policy simile alle seguente sull'endpoint `com.amazonaws.us-east-2.codecommit` che nega specificamente operazioni push Git:

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

```

    },
    {
      "Action": "codecommit:GitPush",
      "Effect": "Deny",
      "Resource": "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
      "Principal": "*"
    }
  ]
}

```

Important

La chiave global condition non `aws:VpcSourceIp` è un repository supportato nelle politiche IAM per i comandi. `CodeCommit git push`

Per ulteriori informazioni, consulta [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC.

Quote in AWS CodeCommit

La tabella seguente descrive le quote in. CodeCommit Per informazioni sulle quote che possono essere modificate, consulta [Endpoint e quote per AWS CodeCommit](#). Per informazioni sulla richiesta di un aumento della quota di servizio, vedere [AWS Service Quotas](#). Per informazioni sulle versioni richieste di Git e altri software, consulta [Compatibilità per CodeCommit, Git e altri componenti](#).

Nomi dei modelli di regola di approvazione e regole di approvazione	Qualsiasi combinazione di lettere, numeri, punti, spazi, caratteri di sottolineatura e trattini, per una lunghezza compresa tra 1 e 100 caratteri. I nomi rispettano la distinzione tra lettere maiuscole e minuscole. I nomi non possono terminare con <code>.git</code> e non possono contenere i seguenti caratteri: <code>! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :</code>
Lunghezza contenuto della regola di approvazione	3000 caratteri

Lunghezza della descrizione del modello di regola di approvazione	1000 caratteri
Riferimenti di destinazione del modello di regola di approvazione	100
Modelli di regole di approvazione	1000 in un Regione AWS
Regole di approvazione su una richiesta pull	Fino a un massimo di 30 Fino a 25 di questi possono derivare da modelli di regola di approvazione.
Regole di approvazione su una richiesta pull create da un modello di regola di approvazione	25
Approvazioni su una richiesta pull	200
Approvatori in un pool di approvazioni	50

Nomi dei rami	Qualsiasi combinazione di caratteri consentiti compresa tra 1 e 256 caratteri di lunghezza, ad eccezione dei nomi di rami composti esattamente da 40 caratteri esadecimali non sono consentiti. I nomi dei rami non possono: • iniziare né terminare con una barra (/) o un punto (.) • contenere il singolo carattere @ • contenere due o più punti consecutivi (.), barre (//) o la seguente combinazione di caratteri: @{ • contenere spazi o uno qualsiasi dei seguenti caratteri: ? ^ * [\ ~ : I nomi dei rami sono riferimenti. Molte delle restrizioni relative ai nomi dei rami sono basate sullo standard di riferimento Git. Per ulteriori informazioni, consulta Git Internals e git-check-ref-format.
Lunghezza del commento	Massimo 10.240 caratteri.
Dimensione compressa dell'archivio per l'inizializzazione di un repository in CodeCommit AWS CloudFormation	Massimo 20 GB quando compresso.
Dati personalizzati per trigger	Campo stringa che può contenere un massimo di 1.000 caratteri. e non può essere utilizzato per passare parametri dinamici.

Display nella console	Un file o un confronto tra file potrebbero non essere visualizzabili nella console se: • Il file è più grande di 2 MB • Il file contiene più di 25.000 caratteri in una sola riga • Un confronto contiene più di 6.500 righe di differenze totali
Indirizzi e-mail nei commit effettuati nella console	Qualsiasi combinazione di caratteri consentiti, per una lunghezza compresa tra 1 e 256 caratteri. Gli indirizzi e-mail non vengono convalidati.

Percorsi di file	Qualsiasi combinazione di caratteri consentiti, per una lunghezza compresa tra 1 e 4.096 caratteri. I percorsi dei file devono essere costituiti da un nome non ambiguo che specifica il file e dalla posizione esatta del file. Oltre a non poter contenere più di 20 directory, Inoltre, i percorsi di file non possono: • contenere stringhe vuote • rappresentare un percorso file relativo • includere una qualsiasi delle seguenti combinazioni di caratteri: <code>./</code> <code>../</code> <code>//</code> • terminare con una barra o una barra rovesciata I nomi e i percorsi di file devono essere completi. Il nome e il percorso di un file sul tuo computer locale devono essere conformi agli standard del sistema operativo in uso. Quando specifichi il percorso di un file in un CodeCommit repository, usa gli standard per Amazon Linux.
Dimensioni file	Massimo 6 MB per ogni singolo file quando usi la CodeCommit console o APIs il. AWS CLI

Dimensioni blob Git	Massimo 2 GB.  Note Non esiste alcuna restrizione relativa al numero di file o alla dimensione totale di tutti i file in un singolo commit, purché i metadati non superino i 6 MB e un singolo blob non superi i 2 GB.
Visualizzazione di rami in formato grafico in Commit Visualizer (Visualizzatore commit)	35 per pagina. Se sono presenti più di 35 rami in un'unica pagina, il grafico non viene visualizzato.
Metadati per un commit	Massimo 20 MB per i metadati combinati per un commit (ad esempio, la combinazione di informazioni sull'autore, data, elenco di commit principale e messaggi di commit) quando si utilizza la CodeCommit console o. APIs AWS CLI  Note Non vi è alcun limite al numero o alla dimensione totale di tutti i file in un singolo commit, purché i metadati non superino i 6 MB, un singolo file non superi i 6 MB e un singolo blob non superi i 2 GB.
Numero di file in un commit	Massimo 100.
Numero di pull request aperte	Massimo 1.000.

Numero di riferimenti in un singolo push	Massimo 4.000, include operazioni di creazione , eliminazione e aggiornamento. Non esiste tuttavia alcuna restrizione sul numero complessivo di riferimenti nel repository.
Numero di archivi	Massimo 5.000 per account Amazon Web Services. Questa restrizione non può essere modificata. Per ulteriori informazioni, consulta AWS CodeCommit Endpoints and Quotas e Service AWS Quotas .
Numero di trigger in un repository	Massimo di 10.

Regioni

CodeCommit è disponibile nelle versioni seguenti: Regioni AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Parigi)
- Europa (Francoforte)
- Europa (Stoccolma)
- Europa (Milano)
- Africa (Città del Capo)
- Israele (Tel Aviv)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Giacarta)
- Medio Oriente (Emirati Arabi Uniti)
- Asia Pacifico (Seoul)
- Asia Pacifico (Osaka)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Hyderabad)
- Asia Pacifico (Hong Kong)
- Sud America (San Paolo)
- Medio Oriente (Bahrein)
- Canada (Centrale)
- Cina (Pechino)
- Cina (Ningxia)

	• AWS GovCloud (Stati Uniti occidentali) • AWS GovCloud (Stati Uniti orientali) Per ulteriori informazioni, consulta Regioni ed endpoint di connessione Git.
Descrizioni dei repository	Qualsiasi combinazione di caratteri, per una lunghezza compresa tra 0 e 1.000 caratteri. Le descrizioni dei repository sono facoltative.
Nomi dei repository	Qualsiasi combinazione di lettere, numeri, punti, caratteri di sottolineatura e trattini, per una lunghezza compresa tra 1 e 100 caratteri. I nomi rispettano la distinzione tra lettere maiuscole e minuscole. I nomi dei repository non possono terminare con .git e non possono contenere i seguenti caratteri: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
I nomi chiave tag repository	Qualsiasi combinazione di lettere Unicode, numeri, spazi e caratteri consentiti in UTF-8 con una lunghezza compresa tra 1 e 128 caratteri. I caratteri consentiti sono + - = . _ : / @ I nomi delle chiavi di tag devono essere univoci e ogni chiave può avere un solo valore. Un tag non può: • iniziare con aws : • contenere solo spazi • terminare con uno spazio • contenere emojis o uno dei seguenti caratteri : ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;

I valori tag repository	Qualsiasi combinazione di lettere Unicode, numeri, spazi e caratteri consentiti in UTF-8 con una lunghezza compresa tra 1 e 256 caratteri. I caratteri consentiti sono + - = . _ : / @ Una chiave può avere un solo valore, ma molte chiavi possono avere lo stesso valore. Un tag non può: • contenere solo spazi • terminare con uno spazio • contenere emojis o uno dei seguenti caratteri : ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Tag repository	I tag rispettano la distinzione tra maiuscole e minuscole. Numero massimo di 50 per ogni risorsa. Non sono consentiti nomi di tag composti esattamente da 40 caratteri esadecimali.
Nomi di trigger	Qualsiasi combinazione di lettere, numeri, punti, caratteri di sottolineatura e trattini, per una lunghezza compresa tra 1 e 100 caratteri. I nomi di trigger non possono contenere spazi o virgole.
Nomi utente nei commit effettuati nella console	Qualsiasi combinazione di caratteri consentiti, per una lunghezza compresa tra 1 e 1.024 caratteri.

AWS CodeCommit riferimento alla riga di comando

Questo riferimento fornisce informazioni su come utilizzare AWS CLI.

Per installare e configurare AWS CLI

1. Sul computer locale, scarica e installa il AWS CLI. Questo è un prerequisito per interagire con la riga CodeCommit di comando. Ti consigliamo di installare la AWS CLI versione 2. È la versione principale più recente di AWS CLI e supporta tutte le funzionalità più recenti. È l'unica versione di AWS CLI che supporta l'utilizzo di un account root, un accesso federato o credenziali temporanee con `git-remote-codecommit`

Per ulteriori informazioni, vedere [Configurazione con l'interfaccia a riga di AWS comando](#).

Note

CodeCommit funziona solo con AWS CLI le versioni 1.7.38 e successive. Come procedura consigliata, installa o esegui l'aggiornamento AWS CLI all'ultima versione disponibile. Per determinare quale versione di AWS CLI è installata, esegui il `aws --version` comando.

Per aggiornare una versione precedente di AWS CLI alla versione più recente, vedi [Installazione di AWS Command Line Interface](#).

2. Esegui questo comando per verificare che i CodeCommit comandi relativi a AWS CLI siano installati.

```
aws codecommit help
```

Questo comando restituisce un elenco di CodeCommit comandi.

3. Configurare AWS CLI con un profilo utilizzando il `configure` comando, come segue:.

```
aws configure
```

Quando richiesto, specifica la chiave di AWS accesso e la chiave di accesso AWS segreta dell'utente IAM con CodeCommit cui utilizzare. Inoltre, assicurati di specificare Regione AWS dove si trova il repository, ad esempio. `us-east-2` Quando viene richiesto il formato di output predefinito, specificare `json`. Ad esempio, se stai configurando un profilo per un utente 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*

Per ulteriori informazioni sulla creazione e la configurazione dei profili da utilizzare con AWS CLI, consulta quanto segue:

- [Profili denominati](#)
- [Utilizzo di un ruolo IAM in AWS CLI](#)
- [Impostazione del comando](#)
- [Connessione ai AWS CodeCommit repository con credenziali rotanti](#)

Per connetterti a un repository o a una risorsa in un altro Regione AWS, devi riconfigurarla AWS CLI con il nome predefinito della regione. I nomi di regione predefiniti supportati includono:

CodeCommit

- 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

Per ulteriori informazioni su CodeCommit e Regione AWS, vedere [Regioni ed endpoint di connessione Git](#). Per ulteriori informazioni su IAM, chiavi di accesso e chiavi segrete, consulta [How Do I Get Credentials?](#) e [gestione delle chiavi di accesso per gli utenti IAM](#). Per ulteriori informazioni sui profili AWS CLI and, consulta [Named Profiles](#).

Per visualizzare un elenco di tutti i CodeCommit comandi disponibili, esegui il comando seguente:

```
aws codecommit help
```

Per visualizzare informazioni su un CodeCommit comando, esegui il comando seguente, *command-name* dov'è il nome del comando (ad esempio, create-repository):

```
aws codecommit command-name help
```

Consulta quanto segue per visualizzare descrizioni ed esempi di utilizzo dei comandi in AWS CLI:

- [associate-approval-rule-template-con-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](#)
- [crea ramo](#)
- [create-commit](#)
- [create-pull-request](#)
- [create-pull-request-approval-regola](#)
- [crea un repository](#)
- [create-unreferenced-merge-commit](#)
- [delete-approval-rule-template](#)
- [delete-branch](#)
- [delete-comment-content](#)
- [delete-file](#)
- [elimina-repository](#)
- [describe-merge-conflicts](#)
- [delete-pull-request-approval-regola](#)
- [describe-pull-request-events](#)
- [disassociate-pull-request-approval-rule-template-from-repository](#)
- [evaluate-pull-request-approval-regole](#)
- [get-approval-rule-template](#)
- [get-blob](#)
- [ottieni una filiale](#)
- [get-comment](#)
- [get-comment-reactions](#)
- [get-comments-for-compared-impegnarsi](#)
- [get-comments-for-pull-richiesta](#)
- [get-commit](#)
- [get-differences](#)
- [get-merge-commit](#)
- [get-merge-conflicts](#)

- [get-merge-options](#)
- [get-pull-request](#)
- [get-pull-request-approval-status](#)
- [get-pull-request-override-status](#)
- [get-repository](#)
- [get-repository-triggers](#)
- [list-approval-rule-templates](#)
- [list-associated-approval-rule-templates-for-repository](#)
- [elenca-rami](#)
- [list-pull-requests](#)
- [elenchi-repository](#)
- [list-repositories-for-approval-modello di regola](#)
- [list-tags-for-resource](#)
- [merge-branches-by-fast-forward](#)
- [merge-branches-by-squash](#)
- [merge-branches-by-three-way](#)
- [merge-pull-request-by-avanzamento rapido](#)
- [merge-pull-request-by-zucca](#)
- [merge-pull-request-by-a tre vie](#)
- [override-pull-request-approval-regole](#)
- [post-comment-for-compared-impegnarsi](#)
- [post-comment-for-pull-richiesta](#)
- [post-comment-reply](#)
- [put-comment-reaction](#)
- [put-file](#)
- [put-repository-triggers](#)
- [tag-resource](#)
- [test-repository-triggers](#)
- [untag-resource](#)
- [update-approval-rule-template-contenuto](#)

- [update-approval-rule-template-descrizione](#)
- [update-approval-rule-template-nome](#)
- [update-comment](#)
- [update-default-branch](#)
- [update-pull-request-approval-contenuto-regola](#)
- [update-pull-request-approval-stato](#)
- [update-pull-request-description](#)
- [update-pull-request-status](#)
- [update-pull-request-title](#)
- [update-repository-description](#)
- [update-repository-name](#)

Comandi Git di base

Puoi usare Git per lavorare con un repository locale e il CodeCommit repository a cui hai collegato il repository locale.

Di seguito sono riportati alcuni esempi di base dei comandi Git utilizzati di frequente.

Per altre opzioni, consulta la documentazione di Git.

Argomenti

- [Variabili di configurazione](#)
- [Archivi remoti](#)
- [Commit](#)
- [Rami](#)
- [Tag](#)

Variabili di configurazione

Elenca tutte le variabili di configurazione.

```
git config --list
```

Elenca solo le variabili di configurazione locali.

```
git config --local -l
```

Elenca solo le variabili di configurazione di sistema.	<code>git config --system -l</code>
Elenca solo le variabili di configurazione globali.	<code>git config --global -l</code>
Imposta una variabile di configurazione nel file di configurazione specificato.	<code>git config [--local --global --system] <i>variable-name</i> <i>variable-value</i></code>
Imposta il nome del ramo predefinito su main per tutti i repository locali quando viene effettuato un commit iniziale su un repository che non ha ancora un ramo predefinito	<code>git config --global init.defaultBranch main</code>
Modifica un file di configurazione direttamente. Può anche essere utilizzato per rilevare il percorso di un file di configurazione specifico. Per uscire dalla modalità di modifica, in genere si digita :q (per uscire senza salvare le modifiche) o :wq (per salvare le modifiche e uscire) e si preme Invio.	<code>git config [--local --global --system] --edit</code>

Archivi remoti

Inizializza un repository locale in preparazione del collegamento a un repository. CodeCommit	<code>git init</code>
Può essere utilizzato per configurare una connessione tra un repository locale e un repository remoto (ad esempio un repository) utilizzando il nickname specificato che il CodeCommit repository locale ha per il repository e l'URL specificato per il CodeCommit repository. CodeCommit	<code>git remote add <i>remote-name</i> <i>remote-url</i></code>
Crea un repository locale creando una copia di un CodeCommit repository all'URL specifica	<code>git clone <i>remote-url</i> <i>local-subfolder-name</i></code>

to, nella sottocartella specificata della cartella corrente sul computer locale. Questo comando crea anche un ramo di tracciamento remoto per ogni ramo nel CodeCommit repository clonato e crea ed estrae un ramo iniziale che deriva dal ramo predefinito corrente nel repository clonato. CodeCommit

Mostra il nickname usato dal repository locale per il repository. CodeCommit

```
git remote
```

Mostra il nickname e l'URL che il repository locale utilizza per i recuperi e i push nel repository. CodeCommit

```
git remote -v
```

Invia i commit finalizzati dal repository locale al repository, utilizzando il nickname specificato che il CodeCommit repository locale ha per il repository e il ramo specificato. CodeCommit Imposta inoltre le informazioni di tracciamento a monte per il repository locale durante il push.

```
git push -u remote-name branch-name
```

Invia i commit finalizzati dal repository locale al CodeCommit repository dopo aver impostato le informazioni di tracciamento upstream.

```
git push
```

Estrae i commit finalizzati al repository locale dal repository, utilizzando il soprannome specificato che il CodeCommit repository locale ha per il repository e il ramo specificato CodeCommit

```
git pull remote-name branch-name
```

Recupera i commit finalizzati al repository locale dal repository dopo aver impostato le informazioni di tracciamento upstream. CodeCommit

```
git pull
```

Disconnette il repository locale dal repository, utilizzando il nickname specificato che il CodeCommit repository locale ha per il repository. CodeCommit

```
git remote rm remote-name
```

Commit

Mostra ciò che è stato o non è stato aggiunto al commit in sospeso nel repository locale.

```
git status
```

Mostra ciò che è stato aggiunto o non è stato aggiunto al commit in sospeso nel repository locale in un formato conciso.

```
git status -sb
```

(M = modificati, A = aggiunti, D = eliminati e così via)

Mostra le modifiche tra il commit in sospeso e il commit più recente nel repository locale.

```
git diff HEAD
```

Aggiunge file specifici al commit in sospeso nel repository locale.

```
git add [file-name-1 file-name-2  
file-name-N | file-pattern ]
```

Aggiunge tutti i file nuovi, modificati ed eliminati al commit in sospeso nel repository locale.

```
git add
```

Inizia a finalizzare il commit in sospeso nel repository locale, che visualizza un editor per fornire un messaggio di commit. Dopo l'immissione del messaggio, il commit in sospeso viene finalizzato.

```
git commit
```

Finalizza il commit in sospeso nel repository locale, inclusa la specificazione di un messaggio di commit contemporaneamente.

```
git commit -m "Some meaningful  
commit comment"
```

Elenca i commit recenti nel repository locale.

```
git log
```

Elenca i commit recenti nel repository locale in formato grafico.	<code>git log --graph</code>
Elenca i commit recenti nel repository locale in un formato condensato predefinito.	<code>git log --pretty=oneline</code>
Elenca i commit recenti nel repository locale in un formato condensato predefinito, con un grafico.	<code>git log --graph --pretty=oneline</code>
Elenca i commit recenti nel repository locale in un formato personalizzato, con un grafico. (Per ulteriori opzioni, vedi Understanding History: Commit nella documentazione di Git.)	<code>git log --graph --pretty=format:"%H (%h) : %cn : %ar : %s"</code>

Rami

Elenca tutti i rami del repository locale con un asterisco (*) visualizzato accanto al ramo corrente.	<code>git branch</code>
Trasferisce le informazioni su tutti i rami esistenti nel CodeCommit repository al repository locale.	<code>git fetch</code>
Elenca tutte le filiali nel repository locale e le filiali di tracciamento remoto nel repository locale.	<code>git branch -a</code>
Elenca solo le filiali di tracciamento remoto nel repository locale.	<code>git branch -r</code>
Crea un nuovo ramo nel repository locale utilizzando il nome del ramo specificato.	<code>git branch <i>new-branch-name</i></code>

Passa a un altro ramo nel repository locale utilizzando il nome del ramo specificato.	<code>git checkout <i>other-branch-name</i></code>
Crea un nuovo ramo nel repository locale utilizzando il nome del ramo specificato, quindi passa ad esso.	<code>git checkout -b <i>new-branch-name</i></code>
Invia un nuovo ramo dal repository locale al repository utilizzando il soprannome specificato che il CodeCommit repository locale ha per il CodeCommit repository e il nome del ramo specificato. Imposta inoltre le informazioni di tracciamento a monte per il ramo nel repository locale durante il push.	<code>git push -u <i>remote-name new-branch-name</i></code>
Crea un nuovo ramo nel repository locale utilizzando il nome del ramo specificato. Quindi collega il nuovo ramo nel repository locale a un ramo esistente nel CodeCommit repository, utilizzando il soprannome specificato che il repository locale ha per il CodeCommit repository e il nome del ramo specificato.	<code>git branch --track <i>new-branch-name remote-name /remote-branch-name</i></code>
Unisce le modifiche da un altro ramo nel repository locale al ramo corrente nel repository locale.	<code>git merge <i>from-other-branch-name</i></code>
Elimina un ramo nel repository locale a meno che non contenga lavori che non sono stati uniti.	<code>git branch -d <i>branch-name</i></code>
Elimina un ramo nel CodeCommit repository utilizzando il soprannome specificato che il repository locale ha assegnato al CodeCommit repository e il nome del ramo specificato. Da notare l'utilizzo dei due punti (:).	<code>git push <i>remote-name :branch-name</i></code>

Tag

Elenca tutti i tag nel repository locale.	<code>git tag</code>
Estrae tutti i tag dal CodeCommit repository al repository locale.	<code>git fetch --tags</code>
Mostra informazioni su un tag specifico nel repository locale.	<code>git show <i>tag-name</i></code>
Crea un tag «leggero» nel repository locale.	<code>git tag <i>tag-name</i> <i>commit-id-to-point-tag-at</i></code>
Invia un tag specifico dal repository locale al repository utilizzando il soprannome specificato che il CodeCommit repository locale ha per il CodeCommit repository e il nome del tag specificato.	<code>git push <i>remote-name</i> <i>tag-name</i></code>
Invia tutti i tag dal repository locale al repository utilizzando il soprannome specificato che il CodeCommit repository locale ha per il repository. CodeCommit	<code>git push <i>remote-name</i> --tags</code>
Elimina un tag nel repository locale.	<code>git tag -d <i>tag-name</i></code>
Elimina un tag nel CodeCommit repository utilizzando il soprannome specificato che il repository locale ha per il CodeCommit repository e il nome del tag specificato. Da notare l'utilizzo dei due punti (:).	<code>git push <i>remote-name</i> :<i>tag-name</i></code>

AWS CodeCommit Cronologia dei documenti della Guida dell'utente

La tabella seguente descrive le modifiche importanti alla documentazione per. CodeCommit Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile iscriversi a un feed RSS.

- Versione API: 2015-04-13

Modifica	Descrizione	Data
AWS CodeCommit è ora disponibile per i nuovi clienti.	AWS CodeCommit è ora disponibile per i nuovi clienti. I clienti esistenti di AWS CodeCommit possono continuare a utilizzare il servizio come di consueto.	25 novembre 2025
AWS CodeCommit non è più disponibile per i nuovi clienti.	AWS CodeCommit non è più disponibile per i nuovi clienti. I clienti esistenti di AWS CodeCommit possono continuare a utilizzare il servizio come di consueto. Scopri di più»	25 luglio 2024
Informazioni aggiornate sulla risoluzione dei problemi per l'helper delle credenziali	Sono state aggiunte informazioni sulla risoluzione degli errori causati da file.gitconfig incompleti o mal configurati.	9 luglio 2024
Informazioni aggiornate sulla risoluzione dei problemi per git-remote-codecommit	Sono state aggiunte informazioni aggiuntive sulla risoluzione degli errori con ambienti gestiti esternamente.	9 luglio 2024

[Rimossi i riferimenti a AWS Toolkit for Eclipse](#)

Ha AWS Toolkit for Eclipse raggiunto la fine del ciclo di vita e non è più supportato. I riferimenti a questo toolkit sono stati rimossi.

9 luglio 2024

[CodeCommit ora supporta l'utilizzo di chiavi gestite dal cliente](#)

Ora puoi utilizzare una chiave gestita dal cliente o una Chiave gestita da AWS per crittografare e decrittografare i dati in un repository. [Per ulteriori informazioni, vedere Crittografia, Creare un repository AWS KMS e Modificare le impostazioni del repository.](#)

21 dicembre 2023

[CodeCommit è ora disponibile in Israele \(Tel Aviv\);](#)

CodeCommit è ora disponibile in Israele (Tel Aviv). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni.](#)

28 agosto 2023

[Modifiche alle politiche gestite per CodeCommit](#)

Le AWSCodeCommitFullAccess politiche AWSCodeCommitPowerUser e sono state aggiornate con un'autorizzazione aggiuntiva. Per ulteriori informazioni, consulta [CodeCommit gli aggiornamenti delle politiche AWS gestite.](#)

16 maggio 2023

[CodeCommit è ora disponibile in tre versioni aggiuntive Regioni AWS](#)

CodeCommit è ora disponibile in altre tre regioni Regioni AWS: Asia Pacifico (Giacarta), Medio Oriente (Emirati Arabi Uniti) e Asia Pacifico (Hyderabad). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni](#).

28 febbraio 2023

[CodeCommit è ora disponibile in Africa \(Città del Capo\)](#)

CodeCommit è ora disponibile in un'altra versione Regione AWS: Africa (Città del Capo). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni](#).

15 settembre 2021

[Modifiche alle politiche gestite per CodeCommit](#)

I dettagli sugli aggiornamenti delle politiche AWS gestite per CodeCommit sono ora disponibili. Per ulteriori informazioni, consulta [CodeCommit gli aggiornamenti delle politiche AWS gestite](#).

18 agosto 2021

[CodeCommit è ora disponibile in Asia Pacifico \(Osaka\)](#)

CodeCommit è ora disponibile in un'altra regione Regione AWS: Asia Pacifico (Osaka). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni](#).

14 aprile 2021

[AWS CloudFormation e AWS Cloud Development Kit \(AWS CDK\) modifica il comportamento di denominazione per le filiali predefinite in CodeCommit](#)

I repository creati utilizzano CloudFormation o AWS CDK con un commit iniziale di codice ora utilizzano il nome del ramo predefinito main. Questa modifica non influisce sui repository o sui rami esistenti. I clienti che utilizzano client Git locali per creare i propri commit iniziali hanno un nome di ramo predefinito che segue la configurazione di tali client Git. Per ulteriori informazioni, consulta [Creazione di CodeCommit risorse con AWS CloudFormation](#).

04 marzo 2021

[CodeCommit modifica il comportamento di denominazione per i rami predefiniti](#)

A partire dal 19 gennaio 2021, il nome del ramo predefinito creato da un commit iniziale in un CodeCommit repository è main. Questa modifica non influisce sui repository o sulle filiali esistenti. I clienti che utilizzano client Git locali per creare i propri commit iniziali hanno un nome di ramo predefinito che segue la configurazione di tali client Git. Per ulteriori informazioni, consulta [Lavorare con le filiali](#), [Creare un commit](#) e [Modificare le impostazioni delle filiali](#).

19 gennaio 2021

[CodeCommit è ora disponibile in Europa \(Milano\)](#)

CodeCommit è ora disponibile in un'altra nazione Regione AWS: Europa (Milano). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni](#).

16 settembre 2020

[CodeCommit aggiunge il supporto per le reazioni emoji ai commenti](#)

CodeCommit ora supporta la reazione ai commenti di altri utenti con emoji. Per ulteriori informazioni, consulta [Commenta un commit e Rivedi una pull Request](#).

24 giugno 2020

[CodeCommit ora disponibile in Cina \(Pechino\) e Cina \(Ningxia\)](#)

CodeCommit è ora disponibile in due versioni aggiuntive Regioni AWS: Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni, consulta [Endpoint di connessione Git e regioni](#).

23 aprile 2020

[CodeCommit aggiunge il supporto per git-remote-codecommit](#)

4 marzo 2020

CodeCommit supporta connessioni ai CodeCommit repository tramite HTTPS con `git-remote-codecommit`, un'utilità che modifica Git. Questo è l'approccio consigliato per le connessioni di accesso federato o temporaneo ai repository. CodeCommit Puoi utilizzarlo anche `git-remote-codecommit` con un utente IAM. `git-remote-codecommit` non richiede l'impostazione delle credenziali Git per l'utente. Per ulteriori informazioni, consulta [Passaggi di configurazione per le connessioni HTTPS a AWS CodeCommit with git-remote-codecommit](#).

[CodeCommit supporta i tag di sessione](#)

CodeCommit supporta l'uso di tag di sessione, che sono attributi di coppia chiave-valore che vengono trasmessi quando si assume un ruolo IAM, si utilizzano credenziali temporanee o si federa un utente in (). AWS Security Token Service AWS STS È possibile utilizzare le informazioni fornite in questi tag per semplificare l'identificazione degli utenti che hanno apportato una modifica o causato un evento. Per ulteriori informazioni, consultare e gli argomenti relativi al [monitoraggio CodeCommit](#) e all'[utilizzo dei tag per fornire informazioni sull'identità in CodeCommit](#).

19 dicembre 2019

[CodeCommit è disponibile in Asia Pacifico \(Hong Kong\)](#)

Ora puoi utilizzarlo CodeCommit in Asia Pacifico (Hong Kong). Per ulteriori informazioni, inclusi gli endpoint di connessione Git, consulta [Regions \(Regioni\)](#).

11 dicembre 2019

[CodeCommit supporta
Amazon CodeGuru Reviewer](#)

CodeCommit supporta Amazon CodeGuru Reviewer, un servizio automatizzato di revisione del codice che utilizza l'analisi dei programmi e l'apprendimento automatico per rilevare problemi comuni e consigliare correzioni nel codice Java o Python. Per ulteriori informazioni, consulta [Associare o dissociare un repository con Amazon CodeGuru Reviewer e Working with Pull Requests](#).

3 dicembre 2019

[CodeCommit supporta le
regole di approvazione](#)

È ora possibile utilizzare le regole di approvazione per personalizzare i flussi di lavoro di sviluppo tra repository in modo che i diversi rami abbiano livelli appropriati di approvazioni e controlli per le richieste pull. Per ulteriori informazioni, consulta [Utilizzo dei modelli di regola di approvazione](#) e [Utilizzo delle richieste pull](#).

20 novembre 2019

[CodeCommit supporta le regole di notifica](#)

È ora possibile utilizzare le regole di notifica per informare gli utenti delle modifiche importanti nei repository. La funzionalità di questa funzione sostituisce le notifiche create prima del 5 novembre 2019. Per ulteriori informazioni, consulta [Creazione di una regola di notifica](#).

05 novembre 2019

[CodeCommit è disponibile in Medio Oriente \(Bahrein\)](#)

Ora puoi usarlo CodeCommit in Medio Oriente (Bahrein). Per ulteriori informazioni, inclusi gli endpoint di connessione Git, consulta [Regions \(Regioni\)](#).

30 ottobre 2019

[CodeCommit aggiunge il supporto per il recupero di informazioni su più commit](#)

È possibile ottenere informazioni su più commit utilizzando il batch-get-commits comando in. AWS CLI Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di commit](#).

15 agosto 2019

[CodeCommit è disponibile in Europa \(Stoccolma\)](#)

Ora puoi usarlo CodeCommit in Europa (Stoccolma). Per ulteriori informazioni, inclusi gli endpoint di connessione Git, consulta [Regions \(Regioni\)](#).

31 luglio 2019

[CodeCommit aggiunge il supporto per l'etichettatura dei repository nella console CodeCommit](#)

Ora puoi aggiungere, gestire e rimuovere tag per un repository per aiutarti a gestire AWS le tue risorse dalla console. CodeCommit Per ulteriori informazioni, consulta [Tagging di un repository](#).

2 luglio 2019

[CodeCommit aggiunge il supporto per ulteriori strategie di fusione Git](#)

È ora possibile scegliere tra strategie di unione Git durante l'unione di richieste di pull in CodeCommit. Puoi anche risolvere i conflitti di fusione nella CodeCommit console. Per ulteriori informazioni, consulta l'articolo sull'[utilizzo delle richieste di pull](#).

10 giugno 2019

[CodeCommit è disponibile in AWS GovCloud \(Stati Uniti orientali\)](#)

Ora puoi usarlo CodeCommit in AWS GovCloud (Stati Uniti orientali). Per ulteriori informazioni, inclusi gli endpoint di connessione Git, consulta [Regions \(Regioni\)](#).

31 maggio 2019

[CodeCommit aggiunge il supporto per l'etichettatura dei repository](#)

Ora puoi aggiungere, gestire e rimuovere tag per un repository per aiutarti a gestire le tue risorse. AWS Per ulteriori informazioni, consulta [Tagging di un repository](#).

30 maggio 2019

[Trova risorse nella console](#)

È ora possibile cercare rapidamente le risorse, ad esempio, repository, progetti di compilazione, applicazioni di distribuzione e pipeline. Scegli Go to resource (Vai alla risorsa) o premi il tasto / e digita il nome della risorsa. Per ulteriori informazioni, consulta [Tutorial CodeCommit](#).

14 maggio 2019

[CodeCommit è disponibile in AWS GovCloud \(Stati Uniti occidentali\)](#)

Ora puoi utilizzarlo CodeCommit in AWS GovCloud (Stati Uniti occidentali). Per ulteriori informazioni, inclusi gli endpoint di connessione Git, consulta [Regions \(Regioni\)](#).

18 Aprile 2019

[CodeCommit aggiunge il supporto per gli endpoint Amazon VPC](#)

Ora puoi stabilire una connessione privata tra il tuo VPC e CodeCommit. Per ulteriori informazioni, consulta [Using CodeCommit with Interface VPC Endpoints](#).

7 marzo 2019

[CodeCommit aggiunge una nuova API](#)

CodeCommit ha aggiunto un'API per la creazione di commit. Per ulteriori informazioni, consulta la sezione [Creazione di un commit](#).

20 febbraio 2019

[Aggiornamento dei contenuti](#)

Il contenuto di questa guida è stato aggiornato con correzioni poco importanti e ulteriori indicazioni per la risoluzione dei problemi.

2 gennaio 2019

[Aggiornamento dei contenuti](#)

Il contenuto di questa guida è stato aggiornato per supportare la nuova esperienza con la CodeCommit console.

30 ottobre 2018

[CodeCommit e lo standard federale per l'elaborazione delle informazioni \(FIPS\)](#)

CodeCommit ha aggiunto il supporto per lo standard governativo Federal Information Processing Standard (FIPS) Publication 140-2 in alcune regioni. Per ulteriori informazioni su FIPS e gli endpoint FIPS, consulta la [panoramica sullo standard Federal Information Processing Standard \(FIPS\) 140-2](#). Per ulteriori informazioni sugli endpoint di connessione Git, consulta [Regioni](#).

25 ottobre 2018

[CodeCommit ne aggiunge tre APIs](#)

CodeCommit ne ha aggiunti tre APIs per supportare l'utilizzo dei file. Per ulteriori informazioni sugli endpoint di connessione Git, consulta [Autorizzazioni per azioni su singoli file](#) e [AWS CodeCommit API Reference](#).

27 settembre 2018

[CodeCommit notifica della cronologia della documentazione disponibile tramite feed RSS](#)

Ora puoi ricevere notifiche sugli aggiornamenti della CodeCommit documentazione iscrivendoti a un feed RSS.

29 giugno 2018

Aggiornamenti precedenti

Nella seguente tabella sono descritte le modifiche significative apportate alla documentazione prima del 29 giugno 2018.

Modifica	Descrizione	Data della modifica
Nuovo argomento	L'argomento Limita le spinte e le fusioni alle filiali è stato aggiunto. L'argomento Riferimento per le autorizzazioni CodeCommit è stato aggiornato.	16 maggio 2018
Nuova sezione	La sezione Lavorare con i file nei AWS CodeCommit repository è stata aggiunta. Gli argomenti Riferimento per le autorizzazioni CodeCommit e Iniziare con AWS CodeCommit sono stati aggiornati.	21 febbraio 2018
Nuovo argomento	L'argomento Configura l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli è stato aggiunto.	21 febbraio 2018
Nuovo argomento	L'argomento Integrazione AWS Cloud9 con AWS CodeCommit è stato aggiunto. L' Integrazioni di prodotti e servizi argomento è stato aggiornato con informazioni su. AWS Cloud9	1° dicembre 2017
Nuova sezione	La sezione Lavorare con le richieste pull nei AWS CodeCommit repository è stata aggiunta. La sezione Autenticazione e controllo degli accessi per AWS CodeCommit è stata aggiornata con informazioni sulle autorizzazioni per le richieste pull e commenti. Include anche istruzioni di policy gestite aggiornate.	20 novembre 2017
Argomenti aggiornati	L' Integrazioni di prodotti e servizi argomento è stato aggiornato per includere collegamenti per i clienti che desiderano aggiornare le pipeline esistenti per utilizzarle e Amazon CloudWatch Events per avviare pipeline in risposta alle modifiche in un CodeCommit repository.	11 ottobre 2017

Modifica	Descrizione	Data della modifica
Nuovi argomenti	La sezione Autenticazione e controllo degli accessi per AWS CodeCommit è stata aggiunta. Sostituisce l'argomento relativo al riferimento delle autorizzazioni di accesso.	11 settembre 2017
Argomenti aggiornati	La sezione Gestione dei trigger per un repository è stata aggiornata per riflettere le modifiche nella configurazione del trigger. Argomenti e immagini sono stati aggiornati in tutta la guida per riflettere le modifiche nella barra di navigazione.	29 agosto 2017
Nuovo argomento	L'argomento Utilizzo delle preferenze dell'utente è stato aggiunto. L'argomento Visualizza i dettagli del tag è stato aggiornato. Gli Integrazioni di prodotti e servizi argomenti sono stati aggiornati con informazioni sull'integrazione con Amazon CloudWatch Events.	3 agosto 2017
Nuovi argomenti	Sono stati aggiunti gli Integra Visual Studio con AWS CodeCommit argomenti AWS CodeCommit Integrate Eclipse con.	29 giugno 2017
Argomento aggiornato	CodeCommit è ora disponibile in due regioni aggiuntive: Asia Pacifico (Mumbai) e Canada (Centrale). L'argomento Regioni ed endpoint di connessione Git è stato aggiornato.	29 giugno 2017
Argomento aggiornato	CodeCommit è ora disponibile in quattro regioni aggiuntive: Asia Pacifico (Seoul), Sud America (San Paolo), Stati Uniti occidentali (California settentrionale) ed Europa (Londra). L'argomento Regioni ed endpoint di connessione Git è stato aggiornato.	6 giugno 2017

Modifica	Descrizione	Data della modifica
Argomento aggiornato	CodeCommit è ora disponibile in quattro regioni aggiuntive: Asia Pacifico (Tokyo), Asia Pacifico (Singapore), Asia Pacifico (Sydney) ed Europa (Francoforte). L' Regioni ed endpoint di connessione Git argomento è stato aggiornato per fornire informazioni sugli endpoint di connessione Git e sulle regioni supportate per CodeCommit.	25 maggio 2017
Nuovo argomento	L'argomento Confronta e unisci filiali è stato aggiunto. Il contenuto della sezione Gestione dei rami è stato aggiornato con informazioni sull'utilizzo della console CodeCommit con rami in un repository.	18 maggio 2017
Nuovo argomento	L'argomento Confrontare i commit è stato aggiunto con informazioni sulla comparazione di commit. La struttura della guida per l'utente è stata aggiornata per l'utilizzo con repository , commit e rami .	28 marzo 2017
Argomento aggiornato	L'argomento Visualizzare i dettagli di commit è stato aggiornato con informazioni sulla visualizzazione delle differenze tra un commit e il relativo padre nella console e sull'uso del comando get-differences per visualizzare le differenze tra commit utilizzando l' AWS CLI.	24 gennaio 2017
Nuovo argomento	L' Registrazione delle chiamate AWS CodeCommit API con AWS CloudTrail argomento è stato aggiunto con informazioni sulla registrazione delle connessioni da utilizzare CodeCommit . AWS CloudFormation	11 gennaio 2017
Nuovo argomento	L' Per utenti HTTPS che utilizzano credenziali Git argomento è stato aggiunto con informazioni sulla configurazione delle connessioni per l' CodeCommit utilizzo delle credenziali Git su HTTPS.	22 dicembre 2016

Modifica	Descrizione	Data della modifica
Argomento aggiornato	L' Integrazioni di prodotti e servizi argomento è stato aggiornato per includere informazioni sull'integrazione con AWS CodeBuild.	5 dicembre 2016
Argomento aggiornato	CodeCommit è ora disponibile in un'altra regione, Europa (Irlanda). L' Regioni ed endpoint di connessione Git argomento è stato aggiornato per fornire informazioni sugli endpoint di connessione Git e sulle regioni supportate per CodeCommit.	16 novembre 2016
Argomento aggiornato	CodeCommit è ora disponibile in un'altra regione, Stati Uniti occidentali (Oregon). L' Regioni ed endpoint di connessione Git argomento è stato aggiornato per fornire informazioni sugli endpoint di connessione Git e sulle regioni supportate per CodeCommit.	14 novembre 2016
Nuovo argomento	L' Creare un trigger per una funzione Lambda argomento è stato aggiornato per riflettere la possibilità di creare CodeCommit trigger come parte della creazione della funzione Lambda. Questo processo semplificato semplifica la creazione dei trigger e configura automaticamente il trigger con le autorizzazioni necessarie per CodeCommit richiamare la funzione Lambda. L' Creare un trigger per una funzione Lambda esistente argomento è stato aggiunto per includere informazioni sulla creazione di trigger per le funzioni Lambda esistenti nella CodeCommit console.	19 Ottobre 2016
Nuovo argomento	CodeCommit è ora disponibile in un'altra regione, Stati Uniti orientali (Ohio). L' Regioni ed endpoint di connessione Git argomento è stato aggiunto per fornire informazioni sugli endpoint di connessione Git e sulle regioni supportate per CodeCommit.	17 ottobre 2016

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	L' Integrazioni di prodotti e servizi argomento è stato aggiornato per includere informazioni sull'integrazione con AWS Elastic Beanstalk.	13 Ottobre 2016
Aggiornamento di argomento	L' Integrazioni di prodotti e servizi argomento è stato aggiornato per includere informazioni sull'integrazione con CloudFormation.	6 ottobre 2016
Aggiornamento di argomento	L'argomento Per le connessioni SSH su Windows è stato modificato per fornire linee guida sull'utilizzo di un emulatore Bash per connessioni SSH in Windows anziché della suite di strumenti PuTTY.	29 settembre 2016
Aggiornamento di argomento	Iniziare con CodeCommit Gli argomenti Visualizzare i dettagli di commit e sono stati aggiornati per includere informazioni sul Commit Visualizer nella CodeCommit console. L'argomento Quote è stato aggiornato con l'aumento del numero di riferimenti consentiti in un singolo push.	14 settembre 2016
Aggiornamento di argomento	Iniziare con CodeCommit Gli argomenti Visualizzare i dettagli di commit e sono stati aggiornati per includere informazioni sulla visualizzazione della cronologia dei commit nella console. CodeCommit	28 luglio 2016
Nuovi argomenti	Gli argomenti Migrare un repository Git su AWS CodeCommit e Esegui la migrazione di contenuti locali o senza versioni su AWS CodeCommit sono stati aggiunti.	29 giugno 2016
Aggiornamento di argomento	Aggiornamenti meno importanti sono stati apportati agli argomenti Risoluzione dei problemi e Per le connessioni HTTPS su Windows con l'helper per le credenziali AWS CLI .	22 giugno 2016

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	Gli argomenti Integrazioni di prodotti e servizi e Access Permissions Reference sono stati aggiornati per includere informazioni sull'integrazione con. CodePipeline	18 Aprile 2016
Nuovi argomenti	La sezione Gestione dei trigger per un repository è stata aggiunta. I nuovi argomenti includono degli esempi, tra cui esempi di policy e di codice, per illustrare il modo in cui creare, modificare ed eliminare trigger.	7 marzo 2016
Nuovo argomento	L'argomento Integrazioni di prodotti e servizi è stato aggiunto. Aggiornamenti meno importanti sono stati apportati a Risoluzione dei problemi .	7 marzo 2016
Aggiornamento di argomento	Oltre all'impronta digitale del MD5 server, l'impronta digitale del SHA256 server per CodeCommit è stata aggiunta a e. Per connessioni SSH su Linux, macOS o Unix Per le connessioni SSH su Windows	9 dicembre 2015
Nuovo argomento	L'argomento Sfoggia i file in un repository è stato aggiunto. Nuovi problemi sono stati aggiunti a Risoluzione dei problemi . Miglioramenti e correzioni meno importanti sono stati apportati in tutta la guida per l'utente.	5 Ottobre 2015
Nuovo argomento	L'argomento Per gli utenti SSH che non utilizzano il AWS CLI è stato aggiunto. Gli argomenti nella sezione Configurazione sono stati semplificati. Linee guida sono state fornite per consentire agli utenti di determinare quale procedura seguire per i propri sistemi operativi e i protocolli preferiti.	5 agosto 2015
Aggiornamento di argomento	Chiarimenti ed esempi sono stati aggiunti alla procedura sull'ID della chiave SSH in SSH e Linux, macOS o Unix: configura le chiavi pubbliche e private per Git e CodeCommit e Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit .	24 luglio 2015

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	I passaggi da seguire sono Passaggio 3: configura le chiavi pubbliche e private per Git e CodeCommit stati aggiornati per risolvere un problema relativo a IAM e al salvataggio del file di chiave pubblica.	22 luglio 2015
Aggiornamento di argomento	Risoluzione dei problemi è stato aggiornato con strumenti di navigazione. Sono state aggiunte ulteriori informazioni sulla risoluzione dei problemi relativi al portachiavi delle credenziali.	20 luglio 2015
Aggiornamento di argomento	Ulteriori informazioni sulle AWS Key Management Service autorizzazioni sono state aggiunte agli argomenti AWS KMS e crittografia e agli argomenti di riferimento sulle autorizzazioni di accesso.	17 luglio 2015
Aggiornamento di argomento	Un'altra sezione è stata aggiunta a Risoluzione dei problemi con informazioni sulla risoluzione dei problemi con AWS Key Management Service.	10 luglio 2015
Rilascio iniziale	Questa è la versione iniziale della Guida per l'utente di CodeCommit .	9 luglio 2015

AWS Glossario

Per la AWS terminologia più recente, consultate il [AWS glossario](#) nella sezione Reference. Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.