

Manual do usuário

AWS CodeCommit



Versão da API 2015-04-13

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

AWS CodeCommit: Manual do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que CodeCommit é	1
Apresentando CodeCommit	2
CodeCommit, Git e como escolher o AWS serviço certo para suas necessidades	3
Como CodeCommit funciona?	7
Como é CodeCommit diferente do controle de versão de arquivos no Amazon S3?	9
Como faço para começar com CodeCommit?	9
Como posso saber mais sobre o Git?	9
Configurando	10
Visualizar e gerenciar suas credenciais	10
Configurar usando credenciais do Git	11
Configurar usando outros métodos	11
Compatibilidade para CodeCommit, Git e outros componentes	13
Para usuários de HTTPS usando credenciais do Git	14
Etapa 1: Configuração inicial para CodeCommit	15
Etapa 2: Instalar o Git	16
Etapa 3: criar credenciais do Git para conexões HTTPS com CodeCommit	16
Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório	18
Próximas etapas	20
Para conexões HTTPS com git-remote-codecommit	20
Etapa 0: instalar os pré-requisitos para git-remote-codecommit	21
Etapa 1: Configuração inicial para CodeCommit	22
Etapa 2: instalar git-remote-codecommit	26
Etapa 3: Conectar-se ao CodeCommit console e clonar o repositório	27
Próximas etapas	28
Para conexões a partir de ferramentas de desenvolvimento	29
Integre AWS Cloud9 com AWS CodeCommit	31
Integre o Visual Studio com o AWS CodeCommit	36
Para usuários de SSH que não usam o AWS CLI	37
Etapa 1: associar sua chave pública ao seu usuário do IAM	38
Etapa 2: adicionar CodeCommit à sua configuração SSH	39
Próximas etapas	39
Para conexões SSH no Linux, macOS ou Unix	40
Etapa 1: Configuração inicial para CodeCommit	40
Etapa 2: Instalar o Git	41

Etapa 3: configurar credenciais no Linux, macOS ou Unix	42
Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório	46
Próximas etapas	48
Para conexões SSH no Windows	48
Etapa 1: Configuração inicial para CodeCommit	48
Etapa 2: Instalar o Git	50
Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit	50
Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório	55
Próximas etapas	56
Para conexões HTTPS no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI	56
Etapa 1: Configuração inicial para CodeCommit	57
Etapa 2: Instalar o Git	60
Etapa 3: Configurar o assistente de credenciais	61
Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório	63
Próximas etapas	64
Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais	64
Etapa 1: Configuração inicial para CodeCommit	65
Etapa 2: Instalar o Git	69
Etapa 3: Configurar o assistente de credenciais	70
Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório	72
Próximas etapas	73
Introdução	74
Começando com CodeCommit	74
Pré-requisitos	75
Etapa 1: criar um CodeCommit repositório	76
Etapa 2: adicionar arquivos ao repositório	79
Etapa 3: navegar pelos conteúdos do seu repositório	81
Etapa 4: criar e colaborar em uma solicitação pull	86
Etapa 5: limpar	93
Etapa 6: próximas etapas	94
Começando com o Git e CodeCommit	94
Etapa 1: criar um CodeCommit repositório	95
Etapa 2: criação de um repositório local	97
Etapa 3: criar sua primeira confirmação	99
Etapa 4: enviar sua primeira confirmação via push	100
Etapa 5: compartilhar o CodeCommit repositório e enviar e receber outro commit	101

Etapa 6: criar e compartilhar uma ramificação	103
Etapa 7: criar e compartilhar uma tag	105
Etapa 8: configurar permissões de acesso	106
Etapa 9: limpar	110
Integrações de produtos e serviços	112
Integração com outros serviços da AWS	112
Exemplos de integração da comunidade	120
Publicações no blog	121
Exemplos de código	124
Trabalhar com repositórios	126
Criar um repositório	127
Criar um repositório (console)	128
Criar um repositório (AWS CLI)	129
Conexão a um repositório	132
Pré-requisitos para se conectar a um repositório CodeCommit	132
Conecte-se ao CodeCommit repositório clonando o repositório	133
Conectar um repositório local ao repositório CodeCommit	135
Compartilhar um repositório	136
Escolha do protocolo de conexão para compartilhar com os usuários	138
Criar políticas do IAM para o seu repositório	139
Criar um grupo do IAM para usuários do repositório	140
Compartilhar as informações de conexão com os usuários	141
Configurar notificações para eventos de repositório	142
Usar regras de notificação do repositório	144
Criar uma regra de notificação	144
Alterar ou desabilitar notificações	148
Excluir notificações	149
Marcar um repositório	150
Adicionar uma tag a um repositório	151
Visualizar as tags de um repositório	153
Editar tags de um repositório	155
Remover uma tag de um repositório	157
Gerenciar triggers para um repositório	158
Crie o recurso e adicione permissões para CodeCommit	159
Criar um gatilho para um tópico do Amazon SNS	159
Criar um gatilho para uma função do Lambda	167

Criar um gatilho para uma função do Lambda existente	172
Editar de gatilhos para um repositório	180
Testar gatilhos para um repositório	182
Excluir gatilhos de um repositório	185
Associar ou desassociar um repositório com o Amazon Reviewer CodeGuru	187
Associar um repositório ao Reviewer CodeGuru	190
Desassociar um repositório do Reviewer CodeGuru	190
Visualizar detalhes de repositórios	191
Visualizar de detalhes de repositórios (console)	191
Exibir detalhes CodeCommit do repositório (Git)	192
Exibir detalhes CodeCommit do repositório ()AWS CLI	193
Alteração de configurações de repositório	197
Alteração de configurações de repositório (console)	197
Alterar as configurações AWS CodeCommit do repositório ()AWS CLI	199
Sincronizar alterações entre repositórios	201
Confirmações de push em dois repositórios	203
Configurar o acesso entre contas a um repositório usando funções	207
Acesso ao repositório entre contas: ações para o administrador na AccountA	208
Acesso ao repositório entre contas: ações para o administrador na AccountB	212
Acesso ao repositório entre contas: ações para o usuário do repositório na AccountB	215
Excluir um repositório	220
Excluir um CodeCommit repositório (console)	221
Excluir um repositório local	222
Excluir um CodeCommit repositório ()AWS CLI	222
Trabalhar com arquivos	224
Procurar arquivos em um repositório	225
Navegue em um CodeCommit repositório	226
Criar ou adicionar um arquivo	227
Criar ou fazer upload de um arquivo (console)	227
Adicionar um arquivo (AWS CLI)	229
Adicionar um arquivo (Git)	230
Editar o conteúdo de um arquivo	230
Editar um arquivo (console)	231
Editar ou excluir um arquivo (AWS CLI)	232
Editar um arquivo (Git)	234
Trabalhar com solicitações pull	235

Criar uma solicitação pull	239
Criar uma solicitação pull (console)	239
Criar uma solicitação pull (AWS CLI)	241
Criar uma regra de aprovação	242
Criar uma regra de aprovação de uma solicitação pull (console)	243
Criar uma regra de aprovação de uma solicitação pull (AWS CLI)	246
Visualizar solicitações pull	247
Visualização de solicitações pull (console)	247
Visualizar solicitações pull (AWS CLI)	248
Revisar uma solicitação pull	253
Revisar uma solicitação pull (console)	253
Revisar solicitações pull (AWS CLI)	258
Atualizar uma solicitação pull	263
Atualizar uma solicitação pull (console)	264
Atualizar solicitações pull (AWS CLI)	264
Editar ou excluir uma regra de aprovação	265
Editar ou excluir uma regra de aprovação de uma solicitação pull (console)	266
Editar ou excluir uma regra de aprovação de uma solicitação pull (AWS CLI)	268
Substituir regras de aprovação em uma solicitação pull	269
Substituir regras de aprovação (console)	270
Substituir regras de aprovação (AWS CLI)	270
Mesclar uma solicitação pull	272
Mesclar uma solicitação pull (console)	273
Mesclar uma solicitação pull (AWS CLI)	276
Resolver conflitos em uma solicitação pull	282
Resolver conflitos em uma solicitação pull (console)	283
Resolver conflitos em uma solicitação pull (AWS CLI)	286
Fechar uma solicitação pull	295
Fechar uma solicitação pull (console)	295
Fechar uma solicitação pull (AWS CLI)	296
Trabalhar com modelos de regra de aprovação	298
Criar um modelo de regra de aprovação	300
Criar um modelo de regra de aprovação (console)	300
Criar um modelo de regra de aprovação (AWS CLI)	305
Associar um modelo de regra de aprovação a um repositório	307
Associar um modelo de regra de aprovação (console)	307

Associar um modelo de regra de aprovação (AWS CLI)	308
Gerenciar modelos de regras de aprovação	309
Gerenciar modelos de regras de aprovação (console)	309
Gerenciar modelos de regras de aprovação (AWS CLI)	310
Desassociar um modelo de regra de aprovação	314
Desassociar um modelo de regra de aprovação (console)	315
Desassociar um modelo de regra de aprovação (AWS CLI)	315
Excluir um modelo de regra de aprovação	316
Excluir um modelo de regra de aprovação (console)	317
Excluir um modelo de regra de aprovação (AWS CLI)	317
Trabalhar com confirmações	318
Criar uma confirmação	319
Crie o primeiro commit para um repositório usando o AWS CLI	320
Criar uma confirmação usando um cliente Git	321
Crie um commit usando o AWS CLI	325
Visualizar detalhes de confirmação	328
Navegar pelas confirmações em um repositório	328
Visualizar detalhes de confirmação (AWS CLI)	333
Visualizar detalhes de confirmação (Git)	339
Comparar confirmações	342
Comparar uma confirmação a seu pai	342
Comparação entre dois especificadores de confirmação	344
Comentar sobre uma confirmação	347
Visualizar comentários em uma confirmação em um repositório	347
Adicionar e responder a comentários em uma confirmação em um repositório	348
Visualizar, adicionar, atualizar e responder a comentários (AWS CLI)	353
Criar uma tag do Git	362
Usar o Git para criar uma tag	362
Visualizar detalhes de tags	363
Visualizar detalhes de tags (console)	364
Visualizar detalhes de tags do Git (Git)	365
Excluir uma tag	367
Usar o Git para excluir uma tag do Git	367
Como trabalhar com ramificações	369
Criar uma ramificação	371
Criar uma ramificação (console)	371

Criar uma ramificação (Git)	372
Criar uma ramificação (AWS CLI)	373
Limitar envios via push e mesclagens em ramificações	375
Configure uma política do IAM para limitar envios por push e mesclagens para ramificações	376
Aplicar a política do IAM a um grupo ou perfil do IAM	378
Teste a política	378
Visualizar detalhes de ramificação	379
Visualizar detalhes de ramificação (console)	379
Visualizar detalhes de ramificação (Git)	380
Visualizar detalhes de ramificação (AWS CLI)	381
Comparar e mesclar ramificações	383
Comparar uma ramificação com a ramificação padrão	383
Comparar duas ramificações específicas	383
Mesclar duas ramificações (AWS CLI)	384
Alterar configurações de ramificação	387
Alterar a ramificação padrão (console)	388
Alterar a ramificação padrão (AWS CLI)	388
Excluir uma ramificação	389
Excluir uma ramificação (console)	390
Excluir uma ramificação (AWS CLI)	390
Excluir uma ramificação (Git)	391
Trabalhar com preferências do usuário	393
Migrar para o CodeCommit	394
Migrar um repositório Git para AWS CodeCommit	394
Etapa 0: configuração necessária para acesso ao CodeCommit	395
Etapa 1: criar um CodeCommit repositório	401
Etapa 2: clonar o repositório e enviar para o repositório CodeCommit	403
Etapa 3: Exibir arquivos em CodeCommit	405
Etapa 4: compartilhar o CodeCommit repositório	405
Migre o conteúdo para CodeCommit	408
Etapa 0: configuração necessária para acessar a CodeCommit	409
Etapa 1: criar um CodeCommit repositório	415
Etapa 2: migrar o conteúdo local para o repositório CodeCommit	417
Etapa 3: Exibir arquivos em CodeCommit	418
Etapa 4: compartilhar o CodeCommit repositório	418

Migrar um repositório em incrementos	421
Etapa 0: determinar se a migração será incremental	421
Etapa 1: instalar os pré-requisitos e adicionar o CodeCommit repositório como remoto	422
Etapa 2: criar o script a ser usado na migração em incrementos	424
Etapa 3: Executar o script e migrar incrementalmente para CodeCommit	425
Apêndice: Amostra de script incremental-repo-migration.py	426
Segurança	435
Proteção de dados	435
AWS KMS e criptografia	436
Usar credenciais rotativas	439
Gerenciamento de Identidade e Acesso	444
Público	444
Autenticação com identidades	445
Gerenciar o acesso usando políticas	446
Autenticação e controle de acesso	448
Como AWS CodeCommit funciona com o IAM	515
CodeCommit políticas baseadas em recursos	516
Autorização baseada em CodeCommit tags	516
CodeCommit Funções do IAM	519
Exemplos de políticas baseadas em identidade	520
Solução de problemas	524
Resiliência	526
Segurança da infraestrutura	527
Monitorar o CodeCommit	528
Monitorar eventos do CodeCommit	529
evento referenceCreated	530
evento referenceUpdated	531
evento referenceDeleted	531
evento unreferencedMergeCommitCreated	532
evento commentOnCommitCreated	533
evento commentOnCommitUpdated	534
evento commentOnPullRequestCreated	535
evento commentOnPullRequestUpdated	536
evento pullRequestCreated	536
evento pullRequestSourceBranchUpdated	538
evento pullRequestStatusChanged	539

evento pullRequestMergeStatusUpdated	540
Evento approvalRuleTemplateCreated	541
Evento approvalRuleTemplateUpdated	541
Evento approvalRuleTemplateDeleted	542
Evento approvalRuleTemplateAssociatedWithRepository	543
Evento approvalRuleTemplateDisassociatedWithRepository	544
Evento approvalRuleTemplateBatchAssociatedWithRepositories	545
Evento approvalRuleTemplateBatchDisassociatedFromRepositories	546
Evento pullRequestApprovalRuleCreated	546
Evento pullRequestApprovalRuleDeleted	548
Evento pullRequestApprovalRuleOverridden	549
Evento pullRequestApprovalStateChanged	551
Evento pullRequestApprovalRuleUpdated	553
Evento reactionCreated	554
Evento reactionUpdated	555
Registrando chamadas de AWS CodeCommit API com AWS CloudTrail	556
CodeCommit informações em CloudTrail	556
Entendendo as entradas do arquivo de CodeCommit log	558
AWS CloudFormation recursos	566
CodeCommit e CloudFormation modelos	566
Exemplos de modelos	567
CloudFormation, CodeCommit, e o AWS Cloud Development Kit (AWS CDK)	568
Saiba mais sobre CloudFormation	569
Solução de problemas	570
Solução de problemas com credenciais do Git (HTTPS)	570
Credenciais do Git para AWS CodeCommit: continuo vendo uma solicitação de credenciais quando me conecto ao meu CodeCommit repositório no terminal ou na linha de comando .	571
Credenciais do Git para AWS CodeCommit: Eu configurei as credenciais do Git, mas meu sistema não as está usando	571
Solução de problemas do git-remote-codecommit	572
Eu vejo um erro: git: 'remote-codecommit' não é um comando git	572
Eu vejo um erro: fatal: Não foi possível encontrar o assistente remoto para 'codecommit'	573
Erro de clonagem: não consigo clonar um CodeCommit repositório de um IDE	573
Erro de push ou pull: não consigo enviar ou extrair commits de um IDE para um repositório CodeCommit	573

Erro de instalação: vejo um externally-managed-environment erro quando tento instalar git-remote-codecommit	574
Solução de problemas com conexões SSH	574
Erro de acesso: o upload da chave pública no IAM foi feito com sucesso, mas a conexão falha nos sistemas Linux, macOS ou Unix	575
Erro de acesso: o upload da chave pública no IAM foi feito com sucesso e o SSH foi testado com sucesso, mas a conexão falha em sistemas Windows	576
Desafio de autenticação: a autenticidade do host não pode ser estabelecida ao se conectar a um repositório CodeCommit	577
Erro do IAM: "Formato inválido" ao tentar adicionar uma chave pública ao IAM	584
Preciso acessar CodeCommit repositórios em várias contas da Amazon Web Services com credenciais SSH	584
Git no Windows: o emulador de Bash ou a linha de comando congelam durante a tentativa de se conectar usando SSH	585
O formato de chave pública requer especificação em algumas distribuições do Linux	586
Erro de acesso: chave pública SSH negada ao se conectar a um repositório CodeCommit ..	586
Solução de problemas com o assistente de credenciais (HTTPS)	586
Eu recebo um erro ao executar o comando <code>git config</code> para configurar o assistente de credenciais	587
Vejo um erro ao ler o nome do usuário quando tento clonar um repositório	588
Recebo um erro de comando não encontrado no Windows ao usar o assistente de credenciais	588
Sou solicitado a fornecer um nome de usuário quando me conecto a um repositório CodeCommit	589
Git para macOS: eu configurei o assistente de credenciais com êxito, mas agora o acesso ao meu repositório é negado (403)	589
Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado (403)	593
Solução de problemas com clientes Git	595
Erro do Git: Error: RPC failed; result=56, HTTP code = 200 fatal: The remote end hung up unexpectedly	595
Erro do Git: comandos de atualização de referência em demasia	596
Erro do Git: o envio por HTTPS está quebrado em algumas versões do Git	596
Erro do Git: "gnutls_handshake() failed"	596
Erro do Git: o Git não consegue encontrar o CodeCommit repositório ou não tem permissão para acessar o repositório	597

Git no Windows: No supported authentication methods available (publickey)	597
Solução de problemas com erros de acesso	597
Erro de acesso: sou solicitado a fornecer um nome de usuário e uma senha quando me conecto a um CodeCommit repositório a partir do Windows	598
Erro de acesso: chave pública negada ao se conectar a um CodeCommit repositório	599
Erro de acesso: mensagem “Taxa excedida” ou “429” ao se conectar a um repositório CodeCommit	599
Solução de problemas de erros de configuração	600
Erro de configuração: não é possível configurar AWS CLI credenciais no macOS	600
Solução de problemas de erros de console	601
Erro de acesso: acesso à chave de criptografia negado para um CodeCommit repositório a partir do console ou AWS CLI	599
Erro de criptografia: o repositório não pode ser descriptografado	601
Erro do console: não é possível pesquisar o código em um CodeCommit repositório a partir do console	602
Erro de exibição: não é possível visualizar um arquivo ou uma comparação entre arquivos	602
Solução de problemas com gatilhos	603
Erro de gatilho: um gatilho do repositório não é executado quando esperado	603
Ativação da depuração	603
CodeCommit referência	605
Regiões e endpoints de conexão do Git	605
Compatível Regiões da AWS com CodeCommit	606
Endpoints de conexão do Git	607
Impressões digitais do servidor para CodeCommit	614
Usando AWS CodeCommit com interface VPC endpoints	621
Disponibilidade	622
Crie endpoints VPC para CodeCommit	623
Crie uma política de VPC endpoint para CodeCommit	624
Cotas	625
Referência da linha de comando	634
Comandos básicos do Git	640
Variáveis de configuração	640
Repositórios remotos	641
Confirmações	643
Ramificações	644

Tags	646
Histórico do documento	647
Atualizações anteriores	659
Glossário da AWS	666
.....	dclxvii

O que AWS CodeCommité

AWS CodeCommit é um serviço de controle de versão hospedado pela Amazon Web Services que você pode usar para armazenar e gerenciar de forma privada ativos (como documentos, código-fonte e arquivos binários) na nuvem. Para obter informações sobre preços de CodeCommit, consulte [Preços](#).

Note

CodeCommit está no escopo de muitos programas de conformidade. Para obter detalhes AWS e esforços de conformidade, consulte [AWS Serviços no escopo do programa de conformidade](#).

Este é um serviço qualificado da HIPAA. [Para obter mais informações sobre AWS a Lei de Portabilidade e Responsabilidade de Seguros de Saúde dos EUA de 1996 \(HIPAA\) e o uso de AWS serviços para processar, armazenar e transmitir informações de saúde protegidas \(PHI\), consulte Visão geral da HIPAA.](#)

Para obter informações sobre esse serviço e a ISO 27001, uma norma de gerenciamento de segurança que especifica as práticas recomendadas de gerenciamento de segurança, consulte [Visão geral da ISO 27001](#).

Para obter informações sobre este serviço e o Padrão de segurança de dados do setor de cartão de pagamento (Payment Card Industry Data Security Standard, PCI DSS), consulte [Visão geral do PCI DSS](#).

Para obter informações sobre este serviço e a publicação 140-2 do padrão do governo dos EUA da Norma federal de processamento de informações (Federal Information Processing Standard, FIPS) que especifica os requisitos de segurança para módulos de criptografia que protegem informações confidenciais, consulte [Visão geral da publicação 140-2 da FIPS e Endpoints de conexão do Git](#).

Tópicos

- [Apresentando CodeCommit](#)
- [CodeCommit, Git e como escolher o AWS serviço certo para suas necessidades](#)
- [Como CodeCommit funciona?](#)
- [Como é CodeCommit diferente do controle de versão de arquivos no Amazon S3?](#)
- [Como faço para começar com CodeCommit?](#)

- [Como posso saber mais sobre o Git?](#)

Apresentando CodeCommit

CodeCommit é um serviço de controle de fonte gerenciado, seguro e altamente escalável que hospeda repositórios Git privados. CodeCommit elimina a necessidade de você gerenciar seu próprio sistema de controle de origem ou se preocupar com a escalabilidade de sua infraestrutura. Você pode usar CodeCommit para armazenar qualquer coisa, desde código até binários. Ele é compatível com a funcionalidade padrão do Git, então funciona perfeitamente com as suas ferramentas baseadas em Git.

Com CodeCommit, você pode:

- Beneficie-se de um serviço totalmente gerenciado hospedado por AWS. CodeCommit fornece alta disponibilidade e durabilidade do serviço e elimina a sobrecarga administrativa de gerenciar seu próprio hardware e software. Não há hardware para provisionar e escalar nem software de servidor para instalar, configurar e atualizar.
- Armazene seu código com segurança. CodeCommit os repositórios são criptografados tanto em repouso quanto em trânsito.
- Trabalhe de forma colaborativa no código. CodeCommit os repositórios oferecem suporte a pull requests, em que os usuários podem revisar e comentar as alterações de código uns dos outros antes de mesclá-las às ramificações; notificações que enviam e-mails automaticamente aos usuários sobre pull requests e comentários; e muito mais.
- Escale seus projetos de controle de versão com facilidade. Os repositórios do CodeCommit podem ser escalados para atender às suas necessidades de desenvolvimento. O serviço pode lidar com repositórios com um grande número de arquivos ou ramificações, tamanhos grandes de arquivo e longos históricos de revisão.
- Armazene qualquer coisa, a qualquer momento. CodeCommit não tem limite no tamanho dos seus repositórios ou nos tipos de arquivo que você pode armazenar.
- Integre-se com outros serviços AWS e serviços de terceiros. CodeCommit mantém seus repositórios próximos aos seus outros recursos de produção na AWS nuvem, o que ajuda a aumentar a velocidade e a frequência do seu ciclo de vida de desenvolvimento. Ele é integrado ao IAM e pode ser usado com outros AWS serviços e em paralelo com outros repositórios. Para obter mais informações, consulte [Integrações de produtos e serviços ao AWS CodeCommit](#).
- Migrar arquivos de outros repositórios remotos facilmente. Você pode migrar para qualquer CodeCommit repositório baseado em Git.

- Use as ferramentas Git que você já conhece. CodeCommit suporta comandos Git, bem como seus próprios AWS CLI comandos e APIs

CodeCommit, Git e como escolher o AWS serviço certo para suas necessidades

Como um serviço baseado em Git, CodeCommit é adequado à maioria das necessidades de controle de versão. Não há limites arbitrários no tamanho dos arquivos, tipos de arquivo e tamanho do repositório. No entanto, há limitações inerentes ao Git que podem afetar negativamente o desempenho de determinados tipos de operações, especialmente ao longo do tempo. Você pode evitar uma possível degradação do desempenho do CodeCommit repositório evitando usá-lo em casos de uso em que outros AWS serviços são mais adequados à tarefa. Você também pode otimizar o desempenho do Git para repositórios complexos. Aqui estão alguns casos de uso em que o Git, portanto CodeCommit, pode não ser a melhor solução para você ou em que talvez você precise tomar medidas adicionais para otimizar o Git.

Caso de uso	Description	Outros serviços a serem considerados:
Arquivos grandes que são alterados com frequência	O Git usa a codificação delta para armazenar diferenças entre versões de arquivos. Por exemplo, se você alterar algumas palavras em um documento, o Git armazenar á somente essas palavras alteradas. Se você tiver arquivos ou objetos acima de 5 MB com muitas alterações, talvez seja necessário que o Git recrie uma grande cadeia de diferenças delta. Isso pode consumir uma quantidade cada vez maior de recursos computacionais no computador local e à CodeCommit	Para arquivos grandes de versão, considere o Amazon Simple Storage Service (Amazon S3). Para obter mais informações, consulte Usando versionamento no Guia do usuário do Amazon Simple Storage Service.

Caso de uso	Description	Outros serviços a serem considerados:
	medida que esses arquivos crescem com o tempo.	
Banco de dados	Os repositórios Git crescem ao longo do tempo. Como o versionamento controla todas as alterações, qualquer alteração aumentará o tamanho do repositório. Em outras palavras, à medida que você confirmar dados, mesmo se você excluir dados em uma confirmação, os dados serão adicionados a um repositório. Conforme a quantidade de dados a serem processados e transmitidos aumenta ao longo do tempo, o Git fica lento. Isso é especialmente prejudicial para um caso de uso de banco de dados. O Git não foi projetado como um banco de dados.	Para criar e usar um banco de dados com desempenho consistente, independentemente do tamanho, considere o Amazon DynamoDB. Para obter mais informações, consulte o Guia de conceitos básicos do Amazon DynamoDB.

Caso de uso	Description	Outros serviços a serem considerados:
Trilhas de auditoria	Normalmente, as trilhas de auditoria são mantidas por longos períodos de tempo e são geradas continuamente por processos do sistema em uma cadência muito frequente. O Git foi projetado para armazenar com segurança o código-fonte gerado por grupos de desenvolvedores em um ciclo de desenvolvimento. Alterar rapidamente os repositórios que armazenam continuamente as alterações do sistema geradas programaticamente causará a degradação do desempenho ao longo do tempo.	Para armazenar trilhas de auditoria, considere o Amazon Simple Storage Service (Amazon S3). Para auditar a AWS atividade, dependendo do seu caso de uso AWS CloudTrailAWS Config, considere usar a Amazon CloudWatch.

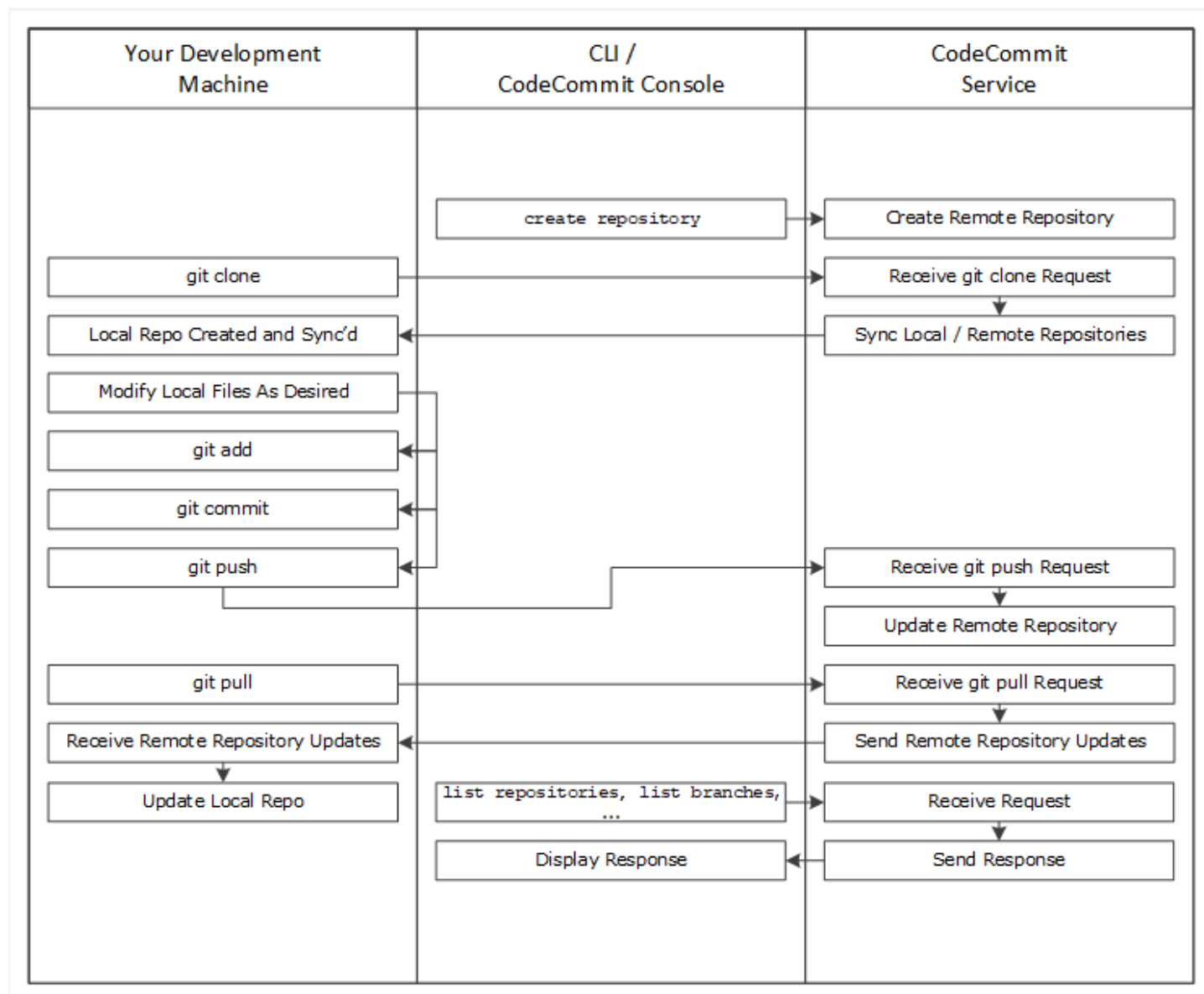
Caso de uso	Description	Outros serviços a serem considerados:
Backups	O Git foi projetado para versionar o código-fonte escrito pelos desenvolvedores. Você pode enviar commits para dois repositórios remotos, incluindo um CodeCommit repositório, como estratégia de backup. No entanto, o Git não foi projetado para lidar com os backups do sistema de arquivos de seu computador, despejos de banco de dados ou conteúdos de backup semelhantes. Isso pode fazer com que o sistema fique lento e aumentar o tempo necessário para clonar e enviar um repositório.	Para obter informações sobre como fazer backup na AWS nuvem, consulte Backup e restauração.

Caso de uso	Description	Outros serviços a serem considerados:
Grande número de ramificações ou referências	Quando um cliente Git envia ou extrai dados do repositório, o servidor remoto deve enviar todas as ramificações e referências, como tags, mesmo que você só esteja interessado em uma única ramificação. Se você tiver milhares de ramificações e referências, poderá levar mais tempo para o processamento e o envio (negociação do pacote) e poderá resultar em resposta aparentemente lenta do repositório. Quanto mais ramificações e tags você tiver, mais esse processo poderá demorar. Recomendamos usar CodeCommit, mas excluir ramificações e tags que não são mais necessárias.	Para analisar o número de referências em um CodeCommit repositório e determinar quais talvez não sejam necessárias, você pode usar um dos seguintes comandos: - Linux, macOS ou Unix, ou emulador Bash no Windows:<pre>git ls-remote wc -l</pre> - Powershell:<pre>git ls-remote Measure-Object -line</pre>

Como CodeCommit funciona?

CodeCommit é familiar para usuários de repositórios baseados em Git, mas mesmo aqueles que não estão familiarizados devem achar a transição para relativamente simples. CodeCommit fornece um console para facilitar a criação de repositórios e a listagem de repositórios e filiais existentes. Em algumas etapas simples, os usuários podem encontrar informações sobre um repositório e cloná-las no computador, criando um repositório local onde podem fazer alterações e enviá-las para o repositório. CodeCommit Os usuários podem trabalhar a partir da linha de comando nas máquinas locais ou usar um editor baseado em GUI.

A figura a seguir mostra como você usa sua máquina de desenvolvimento, o CodeCommit console AWS CLI ou e o CodeCommit serviço para criar e gerenciar repositórios:



1. Use o console AWS CLI ou o CodeCommit console para criar um CodeCommit repositório.
2. Na sua máquina de desenvolvimento, use o Git para executar `git clone`, especificando o nome do repositório. CodeCommit Isso cria um repositório local que se conecta ao CodeCommit repositório.
3. Use o repositório local na sua máquina de desenvolvimento para modificar (adicionar, editar e excluir) arquivos e, então, executar o `git add` para preparar os arquivos modificados localmente. Execute `git commit` para confirmar os arquivos localmente e, em seguida, execute `git push` para enviar os arquivos para o CodeCommit repositório.

4. Baixe alterações de outros usuários. Execute `git pull` para sincronizar os arquivos no CodeCommit repositório com seu repositório local. Isso garante que você trabalhe com a versão mais recente dos arquivos.

Você pode usar o AWS CLI ou o CodeCommit console para rastrear e gerenciar seus repositórios.

Como é CodeCommit diferente do controle de versão de arquivos no Amazon S3?

CodeCommit é otimizado para o desenvolvimento de software em equipe. Ele gerencia lotes de alterações em vários arquivos, que podem ocorrer paralelamente às alterações feitas por outros desenvolvedores. O versionamento do Amazon S3 é compatível com a recuperação de versões anteriores de arquivos, mas não se concentra nos recursos colaborativos de rastreamento de arquivos que as equipes de desenvolvimento de software precisam.

Como faço para começar com CodeCommit?

Para começar com CodeCommit:

1. Siga as etapas em [Configurando](#) para preparar suas máquinas de desenvolvimento.
2. Siga as etapas em um ou mais tutoriais em [Introdução](#).
3. [Crie](#) projetos de controle de versão CodeCommit ou [migre](#) projetos de controle de versão para o CodeCommit

Como posso saber mais sobre o Git?

Se você ainda não souber usar o Git, [deverá aprender a usá-lo](#). Veja alguns recursos úteis:

- [Pro Git](#), uma versão on-line do livro Pro Git. Escrito por Scott Chacon. Publicado pela Apress.
- [Git Immersion](#), um tour try-it-yourself guiado que mostra os fundamentos do uso do Git. Publicado pela Neo Innovation, Inc.
- [Git Reference](#), uma rápida referência on-line que também pode ser usada como um tutorial mais detalhado do Git. Publicado pela GitHub equipe.
- [Git Cheat Sheet](#), com a sintaxe de comando básica do Git. Publicado pela GitHub equipe.
- [Git Pocket Guide](#). Escrito por Richard E. Silverman. Publicado pela O'Reilly Media, Inc.

Configuração do AWS CodeCommit

Você pode fazer login no Console de gerenciamento da AWS e [fazer upload, adicionar ou editar um arquivo](#) em um repositório diretamente no console do AWS CodeCommit. Esta é uma maneira rápida de fazer uma alteração. No entanto, se você quiser trabalhar com vários arquivos, arquivos em ramificações, e assim por diante, considere configurar o computador local para trabalhar com repositórios. A maneira mais fácil de configurar o CodeCommit é configurar as credenciais do Git HTTPS para o AWS CodeCommit. Esse método de autenticação HTTPS:

- Usa um nome de usuário e uma senha estáticos.
- Funciona com todos os sistemas operacionais compatíveis com o CodeCommit.
- É compatível com ambientes de desenvolvimento integrados (IDEs) e outras ferramentas de desenvolvimento com suporte para credenciais do Git.

Você pode usar outros métodos, se não quiser ou não puder usar credenciais do Git por motivos operacionais. Por exemplo, se você acessar repositórios do CodeCommit usando acesso federado, credenciais temporárias ou um provedor de identidade da Web, não poderá usar credenciais do Git. Recomendamos configurar seu computador local usando o comando `git-remote-codecommit`. Revise essas opções cuidadosamente, a fim de decidir qual método alternativo funciona melhor para você.

- [Configurar usando credenciais do Git](#)
- [Configurar usando outros métodos](#)
- [Compatibilidade para CodeCommit, Git e outros componentes](#)

Para obter informações sobre o uso do CodeCommit e da Amazon Virtual Private Cloud, consulte [Usando AWS CodeCommit com interface VPC endpoints](#).

Visualizar e gerenciar suas credenciais

Você pode visualizar e gerenciar suas credenciais do CodeCommit no console da AWS em Minhas credenciais de segurança.

 Note

Essa opção não está disponível para usuários que usam acesso federado, credenciais temporárias ou um provedor de identidade da Web.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM, em <https://console.aws.amazon.com/iam/>.
2. Na barra de navegação no canto superior direito, escolha seu nome de usuário e selecione Credenciais de segurança.
3. Selecione a guia Credenciais do AWS CodeCommit.

Configurar usando credenciais do Git

Com conexões HTTPS e credenciais do Git, você gera um nome de usuário e senha estáticos no IAM. Essas credenciais são usadas com o Git e qualquer outra ferramenta de terceiros com suporte para a autenticação usando o nome de usuário e a senha do Git. Esse método tem suporte da maioria das IDEs e das ferramentas de desenvolvimento. É o método de conexão mais simples e fácil de usar com o CodeCommit.

- [Para usuários de HTTPS usando credenciais do Git](#): siga estas instruções para configurar conexões entre seu computador local e repositórios do CodeCommit usando credenciais do Git.
- [Para conexões a partir de ferramentas de desenvolvimento](#): siga estas orientações para configurar conexões entre seu IDE ou outras ferramentas de desenvolvimento e repositórios do CodeCommit usando credenciais do Git. As IDEs com suporte para credenciais do Git incluem (mas não são limitadas a) Visual Studio, Xcode e IntelliJ.

Configurar usando outros métodos

Você pode usar o protocolo SSH em vez de HTTPS para se conectar ao repositório do CodeCommit. Com conexões SSH, você cria arquivos de chave pública e privada em sua máquina local que o Git e o CodeCommit usam para a autenticação SSH. A chave pública deve ser associada ao seu usuário do IAM. e a chave privada é armazenada na sua máquina local. Como o SSH requer a criação e o gerenciamento manuais de arquivos de chave pública e privada, talvez você ache as credenciais do Git mais simples e fáceis de usar com o CodeCommit.

Diferente das credenciais do Git, a configuração da conexão SSH varia de acordo com o sistema operacional de seu computador local.

- [Para usuários de SSH que não usam o AWS CLI](#): siga estas instruções resumidas se você já tiver um par de chaves pública e privada e estiver familiarizado com conexões SSH em seu computador local.
- [Para conexões SSH no Linux, macOS ou Unix](#): siga essas instruções para obter orientações passo a passo sobre como criar um par de chaves pública e privada e configurar conexões em sistemas operacionais Linux, macOS ou Unix.
- [Para conexões SSH no Windows](#): siga estas instruções para obter orientação passo a passo sobre a criação de um par de chaves pública e privada e a configuração de conexões em sistemas operacionais Windows.

Se você estiver se conectando ao CodeCommit e à AWS usando acesso federado, um provedor de identidade ou credenciais temporárias, ou se não quiser configurar usuários do IAM nem credenciais do Git para usuários do IAM, poderá configurar conexões com repositórios do CodeCommit de duas maneiras:

- Instale e use `git-remote-codecommit` (recomendado).
- Instale e use o assistente de credenciais incluído na AWS CLI.

Com os dois métodos, você pode acessar repositórios do CodeCommit sem a necessidade de um usuário do IAM, ou seja, se conectar a repositórios usando acesso federado e credenciais temporárias. O utilitário `git-remote-codecommit` é a abordagem recomendada. Ele estende o Git e é compatível com uma variedade de versões do Git e assistentes de credenciais. No entanto, nem todos os IDEs oferecem suporte ao formato de URL de clone usado pelo `git-remote-codecommit`. Talvez seja necessário clonar manualmente repositórios no computador local para poder trabalhar com eles no IDE.

- Siga as instruções em [Etapas de configuração para conexões HTTPS a repositórios do AWS CodeCommit com git-remote-codecommit](#) para instalar e configurar `git-remote-codecommit` no Windows, Linux, macOS ou Unix.

O assistente de credenciais incluído na AWS CLI permite que o Git use HTTPS e uma versão assinada com criptografia de suas credenciais de usuário do IAM ou um perfil de instância do Amazon EC2 sempre que ele precisar da autenticação da AWS para interagir com os repositórios

do CodeCommit. Alguns sistemas operacionais e versões do Git têm seus próprios assistentes de credenciais, que entram em conflito com o assistente de credenciais incluído na AWS CLI. Eles podem causar problemas de conectividade no CodeCommit.

- [Para conexões HTTPS no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI](#) : siga estas instruções para obter orientações passo a passo sobre a instalação e a configuração do assistente de credenciais em sistemas Linux, macOS ou Unix.
- [Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais](#): siga estas instruções para obter orientação passo a passo sobre a instalação e a configuração do assistente de credenciais em sistemas Windows.

Se você estiver se conectando a um repositório do CodeCommit hospedado em outra conta da Amazon Web Services, poderá configurar o acesso e definir conexões usando perfis, políticas e o assistente de credenciais incluído na AWS CLI.

- [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#): siga estas instruções para obter uma demonstração passo a passo sobre como configurar o acesso entre contas em uma conta da Amazon Web Services para usuários em um grupo do IAM em outra conta da Amazon Web Services.

Compatibilidade para CodeCommit, Git e outros componentes

Ao trabalhar com o CodeCommit, use o Git. Outros programas também podem ser usados. A tabela a seguir fornece as orientações mais recente sobre compatibilidade de versão. Como prática recomendada, use as versões mais recentes do Git e de outros softwares.

Informações sobre compatibilidade de versões do AWS CodeCommit

Componente	Versão
Git	O CodeCommit é compatível com o Git versões 1.7.9 e posteriores. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Componente	Versão
Curl	O CodeCommit requer o curl 7.33 e versões posteriores. No entanto, há um problema conhecido entre HTTPS e a atualização 7.41.0 do curl. Para obter mais informações, consulte Solução de problemas .
Python (somente git-remote-codecommit)	git-remote-codecommit requer a versão 3 e posterior.
Pip (somente git-remote-codecommit)	git-remote-codecommit requer a versão 9.0.3 e posterior.
AWS CLI (somente git-remote-codecommit)	Recomendamos uma versão recente da AWS CLI versão 2 para todos os usuários do CodeCommit. O git-remote-codecommit requer a AWS CLI versão 2 para ser compatível com o AWS SSO e conexões que exigem credenciais temporárias, como usuários federados.

Configuração para usuários de HTTPS usando credenciais do Git

A maneira mais simples de configurar conexões com AWS CodeCommit repositórios é configurar as credenciais do Git no console do IAM e, em seguida, usar essas credenciais CodeCommit para conexões HTTPS. Você também pode usar essas mesmas credenciais com qualquer ferramenta de terceiros ou ambiente de desenvolvimento integrado (IDE) que seja compatível com autenticação HTTPS usando um nome de usuário e senha estáticos. Para obter exemplos, consulte [Para conexões a partir de ferramentas de desenvolvimento](#).

Note

Se você configurou anteriormente seu computador local para usar o auxiliar de credenciais CodeCommit, você deve editar seu arquivo.gitconfig para remover as informações do auxiliar de credenciais do arquivo antes de poder usar as credenciais do Git. Se o seu computador

local estiver executando macOS, pode ser necessário limpar as credenciais em cachê do Keychain Access.

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para configurar uma conta da Amazon Web Services, criar um usuário do IAM e configurar o acesso CodeCommit a.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#).

Se você quiser usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Recomendamos que você crie um perfil para usar o AWS CLI with CodeCommit. Para obter mais informações, consulte [Referência da linha de comando](#) e [Usar perfis nomeados](#).

Etapa 2: Instalar o Git

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em [Solução de problemas](#).

Etapa 3: criar credenciais do Git para conexões HTTPS com CodeCommit

Depois de instalar o Git, crie credenciais do Git para seu usuário do IAM no IAM.

Para configurar credenciais HTTPS Git para CodeCommit

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

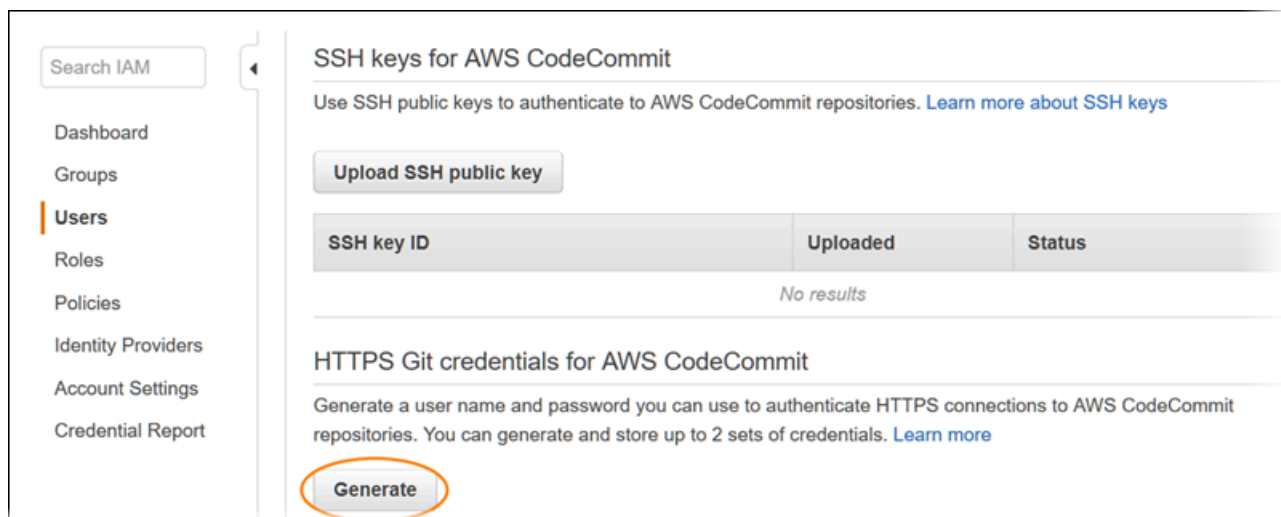
Certifique-se de fazer login como o usuário do IAM que criará e usará as credenciais do Git para conexões com. CodeCommit

2. No painel de navegação do console do IAM, selecione Usuários e, na lista de usuários, escolha o seu usuário do IAM.

Note

Você pode visualizar e gerenciar diretamente suas CodeCommit credenciais em Minhas Credenciais de Segurança. Para obter mais informações, consulte [Visualizar e gerenciar suas credenciais](#).

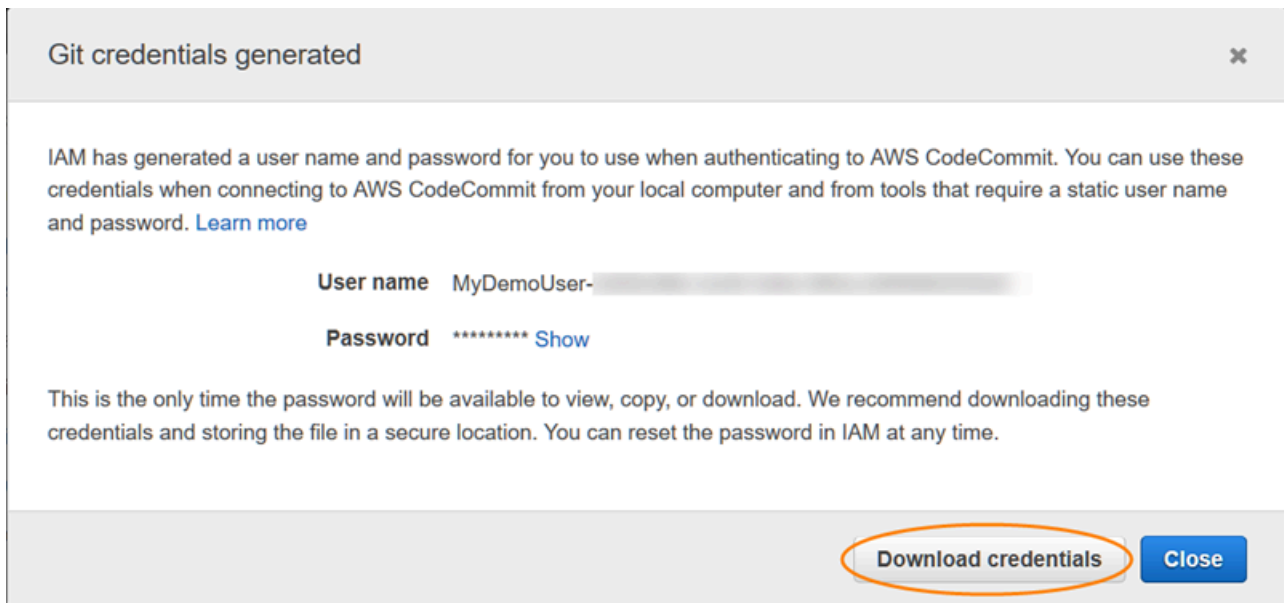
3. Na página de detalhes do usuário, escolha a guia Credenciais de segurança e, em Credenciais HTTPS Git AWS CodeCommit para, escolha Gerar.



Note

Você não pode escolher seu próprio nome do usuário ou senha para as credenciais do Git. Para obter mais informações, consulte [Usar credenciais do Git e HTTPS com. CodeCommit](#)

4. Copie o nome de usuário e a senha que o IAM gerou para você, exibindo, copiando e colando essas informações em um arquivo confiável no seu computador local ou selecionando Fazer download de credenciais para fazer o download dessas informações como um arquivo .CSV. Essas informações são necessárias para você se conectar ao CodeCommit.



Após salvar suas credenciais, selecione Close.

⚠ Important

Essa é a única chance de salvar o nome do usuário e a senha. Se você não salvá-los, poderá copiar o nome de usuário do console do IAM, mas não poderá pesquisar a senha. Você deverá reiniciar a senha e salvá-la.

Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já enviou para você o nome e os detalhes de conexão do repositório do CodeCommit, ignore esta etapa e clone o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.

- Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
- Copie o URL HTTPS (GRC) se estiver usando o comando `git-remote-codecommit` no computador local.
- Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

 Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta na Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. Abra um terminal, uma linha de comando ou um shell do Git. Execute o comando `git clone` com o URL do clone HTTPS copiado para clonar o repositório. Por exemplo, para clonar um repositório com o nome *MyDemoRepo* de um repositório local nomeado *my-demo-repo* na região Leste dos EUA (Ohio):

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

Na primeira vez em que se conectar, será solicitado que você forneça o nome do usuário e a senha do repositório. Dependendo da configuração do computador local, a solicitação pode se originar de um sistema de gerenciamento de credenciais para o sistema operacional, de um utilitário de gerenciamento de credenciais da sua versão do Git (por exemplo, o Git Credential Manager incluído no Git para Windows), do seu IDE ou do próprio Git. Insira o nome de usuário e a senha gerados para as credenciais do Git no IAM (os que você criou em [Etapa 3: criar credenciais do Git para conexões HTTPS com CodeCommit](#)). Dependendo do sistema operacional e de outros softwares, essas informações podem ser salvas para você em um armazenamento de credenciais ou um utilitário de gerenciamento de credenciais. Dessa forma, não deverão mais ser solicitados, a menos que você altere a senha, desative as credenciais do Git ou exclua as credenciais do Git no IAM.

Se você não tiver um armazenamento de credenciais ou um utilitário de gerenciamento de credenciais configurado no seu computador local, instale um. Para obter mais informações sobre o Git e como ele gerencia credenciais, consulte [Credential Storage](#) na documentação do Git.

Para obter mais informações, consulte [Conecte-se ao CodeCommit repositório clonando o repositório](#) e [Criar uma confirmação](#).

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Para saber como criar e enviar via push sua primeira confirmação, consulte [Crie um commit em AWS CodeCommit](#). Se você é novo no Git, revise as informações em [Como posso saber mais sobre o Git?](#) e [Começando com o Git e AWS CodeCommit](#).

Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit

Se você quiser se conectar CodeCommit usando uma conta raiz, acesso federado ou credenciais temporárias, você deve configurar o acesso usando git-remote-codecommit. Esse utilitário fornece um método simples para enviar e extrair código de CodeCommit repositórios estendendo o Git. É o método recomendado para oferecer suporte a conexões feitas com acesso federado, provedores de identidade e credenciais temporárias. Para atribuir permissões a identidades federadas, você cria um perfil e define permissões para o perfil. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidade de terceiros \(federação\)](#) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Guia do Usuário do Centro de Identidade do AWS IAM.

Você também pode usar git-remote-codecommit com um usuário do IAM. Ao contrário de outros métodos de conexão HTTPS, git-remote-codecommit não requer a configuração de credenciais do Git para o usuário.

 Note

Alguns IDEs não suportam o formato de URL de clone usado pelo `git-remote-codecommit`. Talvez seja necessário clonar manualmente repositórios para o computador local para poder trabalhar com eles no IDE de preferência. Para obter mais informações, consulte [Solução de problemas git-remote-codecommit e AWS CodeCommit](#).

Esses procedimentos são escritos com a suposição de que você tem uma conta da Amazon Web Services, criou pelo menos um repositório e usa um usuário do IAM com uma política gerenciada ao se conectar aos CodeCommit repositórios. Para obter informações sobre como configurar o acesso para usuários federados e outros tipos de credenciais rotativas, consulte [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#).

Tópicos

- [Etapa 0: instalar os pré-requisitos para git-remote-codecommit](#)
- [Etapa 1: Configuração inicial para CodeCommit](#)
- [Etapa 2: instalar git-remote-codecommit](#)
- [Etapa 3: Conectar-se ao CodeCommit console e clonar o repositório](#)
- [Próximas etapas](#)

Etapa 0: instalar os pré-requisitos para git-remote-codecommit

Para utilizar `git-remote-codecommit`, é necessário instalar alguns pré-requisitos no computador local. Isso inclui:

- Uma versão compatível do Python. Para obter mais informações sobre as versões compatíveis do Python, consulte [git-remote-codecommit](#)

Para obter mais informações sobre como configurar e usar `git-remote-codecommit`, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

- Git

 Note

Ao instalar o Python no Windows, não se esqueça de escolher a opção para adicionar o Python ao caminho.

git-remote-codecommit requer o pip versão 9.0.3 ou posterior. Para verificar sua versão do pip, abra um terminal ou a linha de comando e execute o seguinte comando:

```
pip --version
```

É possível executar os dois comandos a seguir para atualizar sua versão do pip para a versão mais recente:

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

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

 Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em. [Solução de problemas](#)

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para criar um usuário do IAM, configurá-lo com as políticas apropriadas, obter uma chave de acesso e uma chave secreta e instalar e configurar a AWS CLI.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, verifique se não há políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#)

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. `git-remote-codecommit`

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```

Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, certifique-se de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

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

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

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)
- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

- 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

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Etapa 2: instalar git-remote-codecommit

Siga estas etapas para instalar o git-remote-codecommit.

Para instalar git-remote-codecommit

1. No terminal ou na linha de comando, execute o seguinte comando:

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

Note

Dependendo do sistema operacional e da configuração, talvez seja necessário executar esse comando com permissões elevadas, como sudo, ou usar o parâmetro --user para

instalar em um diretório que não exija privilégios especiais, como sua conta de usuário atual. Por exemplo, em um computador executando Linux, macOS ou Unix:

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

Em um computador executando o Windows:

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

2. Monitore o processo de instalação até que você veja uma mensagem de êxito.

Etapa 3: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já enviou a URL do clone para usar no CodeCommit repositório, você pode pular a conexão com git-remote-codecommit o console e clonar o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.
 - Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
 - Copie o URL HTTPS (GRC) se estiver usando o comando git-remote-codecommit no computador local.
 - Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

 Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta no Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. No terminal ou no prompt de comando, clone o repositório com o comando `git clone`. Use o `git-remote-codecommit` URL HTTPS que você copiou e o nome do AWS CLI perfil, se você criou um perfil nomeado. Se você não especificar um perfil, o comando assumirá o perfil padrão. O repositório local é criado em um subdiretório do diretório onde o comando é executado. Por exemplo, para clonar um repositório chamado *MyDemoRepo* para um repositório local chamado *my-demo-repo*:

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

Para clonar o mesmo repositório usando um perfil chamado: *CodeCommitProfile*

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

Para clonar um repositório em um repositório Região da AWS diferente daquele configurado em seu perfil, inclua o Região da AWS nome. Por exemplo:

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

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Para saber como criar e enviar via push sua primeira confirmação, consulte [Crie um commit em AWS CodeCommit](#). Se você é novo no Git, revise as informações em [Como posso saber mais sobre o Git?](#) e [Começando com o Git e AWS CodeCommit](#).

Configuração de conexões a partir de ferramentas de desenvolvimento usando credenciais do Git

Depois configurar as credenciais do Git para o AWS CodeCommit no console do IAM, você poderá usá-las com qualquer ferramenta de desenvolvimento compatível com credenciais do Git. Por exemplo, você pode configurar o acesso ao repositório do CodeCommit no AWS Cloud9, Visual Studio, Xcode, IntelliJ ou em qualquer ambiente de desenvolvimento integrado (IDE) que inclua as credenciais do Git. Depois de configurar o acesso, você poderá editar seu código, confirmar suas alterações e enviar diretamente do IDE ou de outra ferramenta de desenvolvimento.

Note

Se você acessar repositórios do CodeCommit usando acesso federado, credenciais temporárias ou um provedor de identidade web, não poderá usar credenciais do Git. Recomendamos configurar seu computador local usando o comando `git-remote-codecommit`. No entanto, nem todos os IDEs são totalmente compatíveis com assistentes remotos do Git, como `git-remote-codecommit`. Se você tiver problemas, consulte [Solução de problemas git-remote-codecommit e AWS CodeCommit](#).

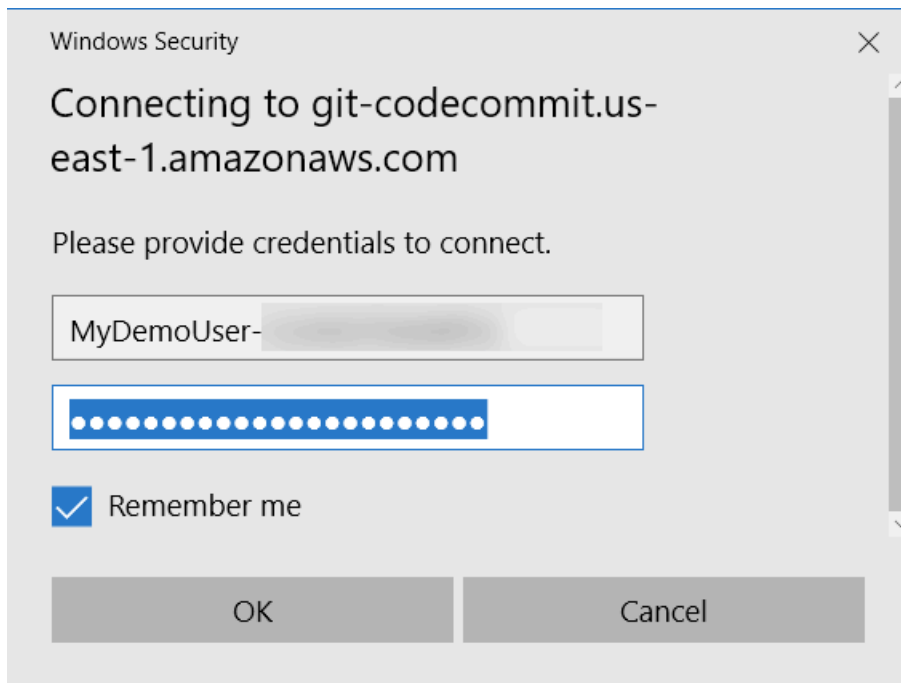
Tópicos

- [Integre AWS Cloud9 com AWS CodeCommit](#)
- [Integre o Visual Studio com o AWS CodeCommit](#)

Quando o IDE ou a ferramenta de desenvolvimento solicitar que você informe o nome de usuário e senha usados para conexão com o repositório do CodeCommit, forneça as credenciais do Git para Nome de usuário e a Senha que você criou no IAM.

Para obter mais informações sobre Regiões da AWS e endpoints do CodeCommit, consulte [Regiões e endpoints de conexão do Git](#).

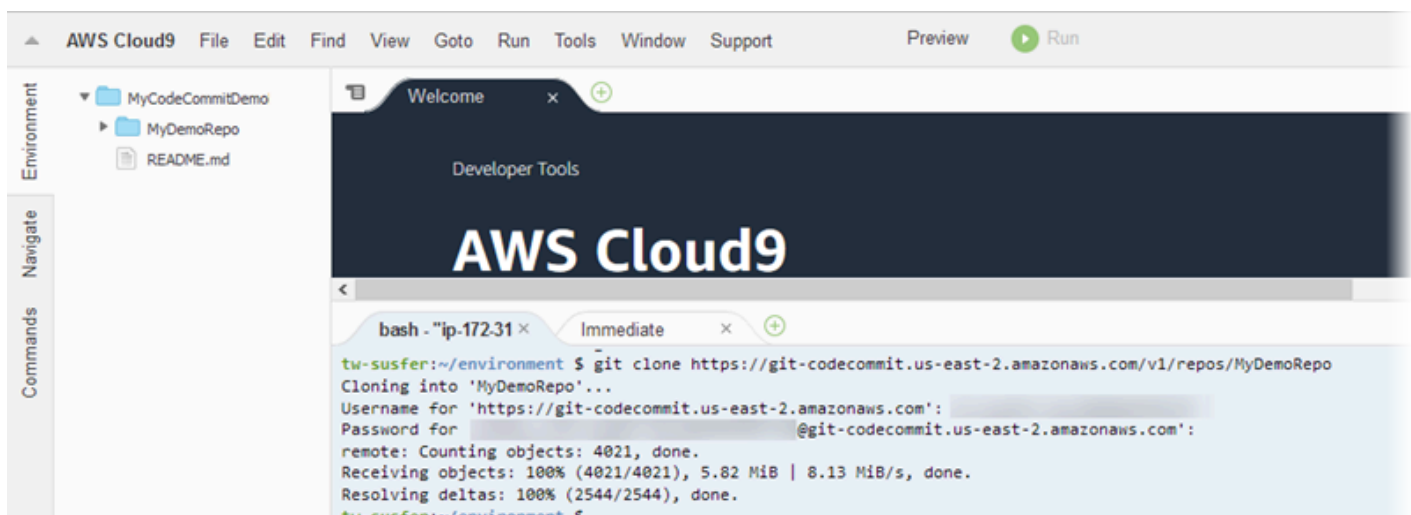
Você também pode ver um prompt do seu sistema operacional para armazenar seu nome de usuário e senha. Por exemplo, no Windows, você forneceria suas credenciais do Git da seguinte maneira:



Para obter informações sobre como configurar credenciais do Git para determinado programa de software ou ferramenta de desenvolvimento, consulte a documentação do produto.

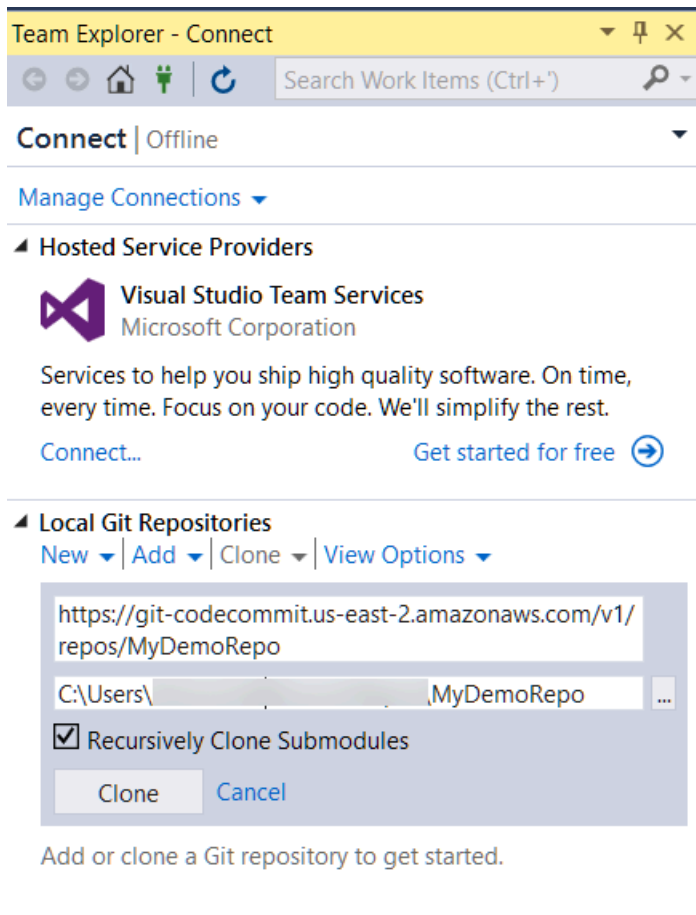
O seguinte não é uma lista abrangente de IDEs. Os links são fornecidos exclusivamente para ajudá-lo a saber mais sobre essas ferramentas. A AWS não se responsabiliza pelo conteúdo de nenhum desses tópicos.

- [AWS Cloud9](#)



- [Visual Studio](#)

Como alternativa, instale o AWS Toolkit for Visual Studio. Para obter mais informações, consulte [Integre o Visual Studio com o AWS CodeCommit](#).



- [XCode](#)

Integre AWS Cloud9 com AWS CodeCommit

Você pode usar AWS Cloud9 para fazer alterações no código em um CodeCommit repositório. AWS Cloud9 contém uma coleção de ferramentas que você pode usar para escrever código e criar, executar, testar, depurar e lançar software. Você pode clonar repositórios existentes, criar repositórios, confirmar e enviar alterações de código para um repositório e muito mais, tudo a partir do seu AWS Cloud9 ambiente de desenvolvimento EC2. O ambiente de desenvolvimento do AWS Cloud9 EC2 geralmente é pré-configurado com a AWS CLI, uma função do Amazon EC2 e o Git. Portanto, na maioria dos casos, você pode executar alguns comandos simples e começar a interagir com seu repositório.

Para usar AWS Cloud9 com CodeCommit, você precisa do seguinte:

- Um ambiente de desenvolvimento AWS Cloud9 EC2 executado no Amazon Linux.
- O AWS Cloud9 IDE é aberto em um navegador da web.
- Um usuário do IAM com uma das políticas CodeCommit gerenciadas e uma das políticas AWS Cloud9 gerenciadas aplicadas a ela.

Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#) e [Compreender os conceitos e suas credenciais de segurança](#).

Note

Este tópico descreve a configuração da integração com CodeCommit e AWS Cloud9 com o acesso geral da Internet. Você pode configurar o acesso a CodeCommit e AWS Cloud9 em um ambiente isolado, mas isso requer etapas adicionais. Para obter mais informações, consulte:

- [Usando AWS CodeCommit com interface VPC endpoints](#)
- [Acessando instâncias sem entrada do Amazon EC2 com AWS Systems Manager](#)
- [Como trabalhar com ambientes compartilhados](#)
- [Compartilhar sua VPC com outras contas](#)
- [Publicação no blog: Isolando o acesso à rede em seus ambientes AWS Cloud9](#)

Tópicos

- [Etapa 1: criar um ambiente AWS Cloud9 de desenvolvimento](#)
- [Etapa 2: Configurar o auxiliar de AWS CLI credenciais em seu ambiente de desenvolvimento AWS Cloud9 EC2](#)
- [Etapa 3: clonar um CodeCommit repositório em seu ambiente de desenvolvimento AWS Cloud9 EC2](#)
- [Próximas etapas](#)

Etapa 1: criar um ambiente AWS Cloud9 de desenvolvimento

AWS Cloud9 hospeda seu ambiente de desenvolvimento em uma instância do Amazon EC2. Essa é a maneira mais fácil de integrar, pois você pode usar as credenciais temporárias AWS gerenciadas da instância para se conectar ao seu CodeCommit repositório. Para usar seu próprio servidor, consulte o [Guia do usuário do AWS Cloud9](#).

Para criar um AWS Cloud9 ambiente

1. Faça login AWS como o usuário do IAM que você configurou e abra o AWS Cloud9 console.
2. No AWS Cloud9 console, escolha Criar ambiente.
3. Em Step 1: Name environment (Etapa 1: nomear ambiente), insira um nome e uma descrição opcional para o ambiente e selecione Next step (Próxima etapa).
4. Em Step 2: Configure Settings, configure seu ambiente da seguinte maneira:
 - Em Environment type, escolha Create a new instance for environment (EC2).
 - Em Instance type, escolha o tipo de instância adequado para o seu ambiente de desenvolvimento. Por exemplo, se você está apenas explorando o serviço, pode escolher o padrão de t2.micro. Se você pretende usar este ambiente para trabalho de desenvolvimento, escolha um tipo de instância maior.
 - Aceite as outras configurações padrão, a menos que você tenha motivos para escolher o contrário (por exemplo, sua organização usa uma VPC específica ou sua conta da Amazon Web Services não tem nenhuma VPCs configurada) e, em seguida, escolha Próxima etapa.
5. Em Step 3: Review, analise suas configurações. Escolha Previous step se quiser fazer qualquer alteração. Se não, escolha Create environment.

A criação de um ambiente e a conexão a ele pela primeira vez leva vários minutos. Se parecer levar um tempo muito longo, consulte [Solução de problemas](#) no Guia do usuário do AWS Cloud9.

6. Após a conexão com o ambiente, verifique se o Git já está instalado e se é uma versão compatível, executando o comando `git --version` na janela do terminal.

Se o Git não estiver instalado, ou se não for uma versão compatível, instale uma versão compatível. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git. Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Tip

Dependendo do sistema operacional do ambiente, você pode usar o comando `yum` com a opção `sudo` para instalar atualizações, incluindo o Git. Por exemplo, uma sequência de comandos administrativos poderá se parecer com os seguintes três comandos:

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

7. Configure um nome de usuário e e-mail a ser associado a suas confirmações do Git executando o comando `git config`. Por exemplo:

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

Etapa 2: Configurar o auxiliar de AWS CLI credenciais em seu ambiente de desenvolvimento AWS Cloud9 EC2

Depois de criar um AWS Cloud9 ambiente, você pode configurar o auxiliar de AWS CLI credenciais para gerenciar as credenciais das conexões ao seu repositório. CodeCommit O ambiente de AWS Cloud9 desenvolvimento vem com credenciais temporárias AWS gerenciadas associadas ao seu usuário do IAM. Você usa essas credenciais com o auxiliar de AWS CLI credenciais.

1. Abra a janela do terminal e execute o seguinte comando para verificar se a AWS CLI está instalada:

```
aws --version
```

Se for bem-sucedido, este comando retornará a versão da AWS CLI instalada no momento. Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. No terminal, execute os seguintes comandos para configurar o auxiliar de AWS CLI credenciais para conexões HTTPS:

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


 Tip

O assistente de credenciais usa o perfil de instância padrão do Amazon EC2 para o ambiente de desenvolvimento. Se você pretende usar o ambiente de desenvolvimento para se conectar a repositórios que não estão hospedados CodeCommit, configure conexões SSH com esses repositórios ou configure um `.gitconfig` arquivo local para usar um sistema alternativo de gerenciamento de credenciais ao se conectar a esses outros repositórios. Para obter mais informações, consulte [Ferramentas do GiT – armazenamento de credenciais](#) no site do Git.

Etapa 3: clonar um CodeCommit repositório em seu ambiente de desenvolvimento AWS Cloud9 EC2

Depois de configurar o auxiliar de AWS CLI credenciais, você pode clonar seu CodeCommit repositório nele. Em seguida, pode começar a trabalhar com o código.

1. No terminal, execute o comando `git clone` especificando a URL do clone HTTPS do repositório que você deseja clonar. Por exemplo, se você quiser clonar um repositório chamado `MyDemoRepo` na região Leste dos EUA (Ohio), insira:

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

 Tip

Você pode encontrar o URL do clone do seu repositório no CodeCommit console escolhendo Clone URL.

2. Quando a clonagem for concluída, na navegação lateral, expanda a pasta do seu repositório e selecione o arquivo que deseja abrir para edição. Como alternativa, selecione File (Arquivo) e selecione New file (Novo arquivo) para criar um arquivo.
3. Quando você finalizar a edição ou a criação de arquivos, na janela do terminal, altere os diretórios para o repositório clonado e, em seguida, confirme e faça push de suas alterações. Por exemplo, se você adicionou um novo arquivo chamado `MyFile.py`:

```
cd MyDemoRepo
git commit -a MyFile.py
```

```
git commit -m "Added a new file with some code improvements"
git push
```

Próximas etapas

Para obter mais informações, consulte o [Guia AWS Cloud9 do usuário](#) e a [CodeCommit amostra de AWS Cloud9](#). Para obter mais informações sobre como usar o Git com CodeCommit, consulte [Começando com o Git e AWS CodeCommit](#).

Integre o Visual Studio com o AWS CodeCommit

Você pode usar o Visual Studio para fazer alterações de código em um repositório do CodeCommit. O AWS Toolkit for Visual Studio agora inclui atributos que tornam o trabalho no CodeCommit mais fácil e prático ao usar Visual Studio. A integração do Toolkit do Visual Studio foi desenvolvida para funcionar com credenciais do Git e usuário do IAM. Você pode clonar repositórios existentes, criar repositórios, confirmar e enviar alterações de código via push para um repositório e muito mais.

Important

O Toolkit for Visual Studio está disponível para instalação apenas em sistemas operacionais Windows. Se você estiver procurando informações sobre como trabalhar com o Visual Studio Code, consulte [AWS Toolkit for Visual Studio Code](#).

Se você usou já usou o Toolkit for Visual Studio, provavelmente já está familiarizado com a configuração de perfis de credenciais da AWS que contêm uma chave de acesso e uma chave secreta. Os perfis de credenciais são usados no Toolkit for Visual Studio para habilitar chamadas às APIs de serviço da AWS (por exemplo, para que o Amazon S3 liste buckets ou para que o CodeCommit liste repositórios). Para extrair e enviar via push códigos a um repositório do CodeCommit, você também precisa de credenciais do Git. Se você não tiver credenciais do Git, o Toolkit for Visual Studio poderá gerar e aplicar essas credenciais para você. Isso pode poupar muito tempo.

Para usar o Visual Studio com o CodeCommit, você precisa do seguinte:

- Um usuário do IAM com um conjunto válido de credenciais (uma chave de acesso e uma chave secreta) configuradas para isso. Esse usuário do IAM também deve ter:

Uma das políticas gerenciadas do CodeCommit e a política gerenciada do `IAMSelfManageServiceSpecificCredentials` aplicadas a ele.

OU

Se o usuário do IAM já tiver credenciais do Git configuradas, uma das políticas gerenciadas do CodeCommit ou permissões equivalentes.

Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#) e [Compreender os conceitos e suas credenciais de segurança](#).

- O AWS Toolkit for Visual Studio instalado no computador no qual você instalou o Visual Studio. Para obter mais informações, consulte [Configurar o AWS Toolkit for Visual Studio](#).

Para obter mais informações sobre como usar o AWS Toolkit for Visual Studio com o CodeCommit, consulte [Usar o AWS CodeCommit com o Visual Studio Team Explorer](#) no Guia do Usuário do Toolkit for Visual Studio.

Configuração para usuários SSH que não usam o AWS CLI

Se quiser usar conexões SSH para seu repositório, você pode se conectar AWS CodeCommit sem instalar o AWS CLI. O AWS CLI inclui comandos que são úteis quando você usa e gerencia CodeCommit repositórios, mas não são necessários para a configuração inicial.

Este tópico supõe que:

- Você configurou um usuário do IAM com as políticas ou permissões necessárias CodeCommit e a política `IAMUserSSHKeys` gerenciada ou permissões equivalentes necessárias para o upload das chaves. Para obter mais informações, consulte [Usando políticas baseadas em identidade \(políticas do IAM\) para CodeCommit](#).
- Você já possui ou sabe criar um par de chaves públicas/privadas. É altamente recomendável que você use uma senha confiável para a sua chave SSH.
- Você está familiarizado com SSH, o cliente do Git e seus arquivos de configuração.
- Se estiver usando o Windows, você tem um utilitário de linha de comando instalado, como o Git Bash, que emula o shell bash.

Se precisar de mais orientações, siga as instruções em [Para conexões SSH no Linux, macOS ou Unix](#) ou [Para conexões SSH no Windows](#).

Tópicos

- Etapa 1: associar sua chave pública ao seu usuário do IAM
- Etapa 2: adicionar CodeCommit à sua configuração SSH
- Próximas etapas

Etapa 1: associar sua chave pública ao seu usuário do IAM

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, selecione Usuários e, na lista de usuários, escolha o seu usuário do IAM.
3. Na guia Security Credentials, selecione Upload SSH public key.
4. Cole os conteúdos da chave SSH pública no campo e selecione Upload SSH Key.

 Tip

Os par de chaves públicas/privadas deve ser SSH-2 RSA, em formato OpenSSH, e conter 2.048 bits. A chave é semelhante a esta:

```
ssh-rsa EXAMPLE-
AfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJB
gNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA
5zZb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxH2AdBgkqhkiG9w0BCQEWEG5vb251QG
FtYXpvi5jb20wHhc
NMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJB
gNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=
EXAMPLE user-
name@ip-192-0-2-137
```

O IAM só aceita chaves públicas em formato OpenSSH. Se você fornecer a chave pública em outro formato, verá uma mensagem de erro informando que o formato da chave é inválido.

5. Copie o ID da chave SSH (por exemplo, *APKAEIBAERJR2EXAMPLE*) e feche o 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

Etapa 2: adicionar CodeCommit à sua configuração SSH

1. No terminal (Linux, macOS ou Unix) ou no emulador de bash (Windows), edite o arquivo de configuração de 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 houver mais de uma configuração de SSH, não se esqueça de incluir as linhas em branco antes e depois do conteúdo. Salve o arquivo pressionando as teclas `Ctrl` e `d` simultaneamente.

2. Execute o seguinte comando para testar a sua configuração de SSH:

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

Insira a senha do arquivo da chave SSH quando solicitado. Se tudo for configurado corretamente, você verá a seguinte mensagem de êxito:

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

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Para se conectar a um repositório, siga as etapas em [Conexão a um repositório](#). Para criar um repositório, siga as etapas em [Criar um repositório](#).

Etapas de configuração para conexões SSH com AWS CodeCommit repositórios no Linux, macOS ou Unix

Antes de se conectar CodeCommit pela primeira vez, você deve concluir algumas etapas iniciais de configuração. Depois de configurar o computador e o AWS perfil, você pode se conectar a um CodeCommit repositório e clonar esse repositório no seu computador (também conhecido como criar um repositório local). Se não estiver familiarizado com o Git, revise as informações em [Como posso saber mais sobre o Git?](#).

Tópicos

- [Etapa 1: Configuração inicial para CodeCommit](#)
- [Etapa 2: Instalar o Git](#)
- [Etapa 3: configurar credenciais no Linux, macOS ou Unix](#)
- [Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório](#)
- [Próximas etapas](#)

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para configurar uma conta da Amazon Web Services, criar um usuário do IAM e configurar o acesso CodeCommit a.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#)

 Note

Se você quiser usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Etapa 2: Instalar o Git

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A

versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

 Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em: [Solução de problemas](#)

Etapa 3: configurar credenciais no Linux, macOS ou Unix

SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas para Git e CodeCommit

Para configurar as chaves públicas e privadas para o Git e CodeCommit

1. No terminal da sua máquina local, execute o comando `ssh-keygen` e siga as orientações para salvar o arquivo no diretório `.ssh` do seu perfil.

 Note

Verifique com o administrador do sistema onde os arquivos de chave devem ser armazenados e qual padrão de nomeação de arquivos deve ser usado.

Por exemplo:

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

```

Isso gera:

- O *codecommit_rsa* arquivo, que é o arquivo de chave privada.
- O *codecommit_rsa.pub* arquivo, que é o arquivo de chave pública.

Tip

Por padrão, o `ssh-keygen` gera uma chave de 2048 bits. Você pode usar os parâmetros `-t` e `-b` para especificar o tipo e o comprimento da chave. Se você quiser uma chave de 4096 bits no formato `rsa`, especifique isso executando o comando com os seguintes parâmetros:

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

Para obter mais informações sobre os formatos e comprimentos necessários para as chaves SSH, consulte Como [usar o IAM com](#). CodeCommit

2. Execute o comando a seguir para exibir o valor do arquivo de chave pública (*codecommit_rsa.pub*):

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

Copie o valor. Ele é similar ao seguinte:

```
ssh-rsa EXAMPLE-AfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWV6b24xZDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQFEWEG5vb251QGftYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWV6b24xZDAS=EXAMPLE user-name@ip-192-0-2-137
```

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

 Note

Você pode visualizar e gerenciar diretamente suas CodeCommit credenciais em Minhas Credenciais de Segurança. Para obter mais informações, consulte [Visualizar e gerenciar suas credenciais](#).

- No painel de navegação do console do IAM, selecione Usuários e, na lista de usuários, escolha o seu usuário do IAM.
- Na página de detalhes do usuário, selecione a guia Security Credentials e, em seguida, Upload SSH public key.
- Cole os conteúdos da chave SSH pública no campo e selecione Upload SSH public key.
- Copie ou salve as informações no ID da chave SSH (por exemplo, **APKAEIBAERJR2EXAMPLE**).

SSH keys for AWS CodeCommit

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

[Upload SSH public key](#)

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

 Note

Se você tiver mais de uma chave SSH IDs carregada, as chaves serão listadas alfabeticamente por ID da chave, não pela data de upload. Verifique se você copiou o ID da chave que está associado à data de upload correta.

8. Em sua máquina local, use um editor de texto para criar um arquivo de configuração no diretório `~/.ssh` e, em seguida, adicione as seguintes linhas ao arquivo, onde o valor para *User* é o ID da chave SSH que você copiou anteriormente:

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

 Note

Se você deu ao seu arquivo de chave privada um nome diferente de *codecommit_rsa*, não se esqueça de usá-lo aqui.

Você pode configurar o acesso SSH aos repositórios em várias contas da Amazon Web Services. Para obter mais informações, consulte [Solução de problemas de conexões SSH com AWS CodeCommit](#).

Salve e nomeie o arquivo como `config`.

9. No terminal, execute o seguinte comando para alterar as permissões do arquivo de configuração:

```
chmod 600 config
```

10. Execute o seguinte comando para testar a sua configuração de SSH:

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

Você será solicitado a confirmar a conexão porque `git-codecommit.us-east-2.amazonaws.com` ainda não está incluído em seu arquivo de hosts conhecidos. A impressão digital CodeCommit do servidor é exibida como parte da verificação (a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e para MD5 ou 31B1W2g5xn/NA2Ck6dyeJIrQ0Wvn7n8UEs56fG6ZIzQ para SHA256).

 Note

CodeCommit as impressões digitais do servidor são exclusivas para cada Região da AWS um. Para ver as impressões digitais do servidor de um Região da AWS, consulte [Impressões digitais do servidor para CodeCommit](#).

Após confirmar a conexão, aparecerão a confirmação de que você adicionou o servidor ao seu arquivo hosts conhecido e uma mensagem de conexão bem-sucedida. Se você não ver uma mensagem de sucesso, verifique se você salvou o config arquivo no diretório ~/.ssh do usuário do IAM ao qual você configurou o acesso e se CodeCommit especificou o arquivo de chave privada correto.

Para obter informações que ajudam na resolução de problemas, execute o comando ssh com o parâmetro -v. Por exemplo:

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

Para obter informações para ajudar a solucionar problemas de conexão, consulte [Solução de problemas de conexões SSH com AWS CodeCommit](#).

Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já enviou para você o nome e os detalhes de conexão do repositório do CodeCommit, ignore esta etapa e clone o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.

- Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
- Copie o URL HTTPS (GRC) se estiver usando o comando `git-remote-codecommit` no computador local.
- Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

 Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta na Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. Abra um terminal. No diretório `/tmp`, execute o comando `git clone` com o URL SSH que você copiou para clonar o repositório. Por exemplo, para clonar um repositório com o nome *MyDemoRepo* de um repositório local nomeado *my-demo-repo* na região Leste dos EUA (Ohio):

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

 Note

Se você testou a sua conexão com êxito, mas o comando do clone falhar, talvez você não tenha o acesso necessário ao arquivo de configuração ou talvez outra configuração esteja em conflito com ele. Tente se conectar novamente e inclua o ID da chave SSH no comando dessa vez. Por exemplo:

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

Para obter mais informações, consulte [Erro de acesso: o upload da chave pública no IAM foi feito com sucesso, mas a conexão falha nos sistemas Linux, macOS ou Unix](#).

Para obter mais informações sobre como conectar-se a repositórios, consulte [Conecte-se ao CodeCommit repositório clonando o repositório](#).

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Etapas de configuração para conexões SSH com AWS CodeCommit repositórios no Windows

Antes de se conectar AWS CodeCommit pela primeira vez, você deve concluir algumas etapas iniciais de configuração. Depois de configurar o computador e o AWS perfil, você pode se conectar a um CodeCommit repositório e clonar esse repositório no seu computador (também conhecido como criar um repositório local). Se não estiver familiarizado com o Git, revise as informações em [Como posso saber mais sobre o Git?](#).

Tópicos

- [Etapa 1: Configuração inicial para CodeCommit](#)
- [Etapa 2: Instalar o Git](#)
- [Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit](#)
- [Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório](#)
- [Próximas etapas](#)

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para configurar uma conta da Amazon Web Services, criar um usuário do IAM e configurar o acesso CodeCommit a.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.

2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório e Autenticação e controle de acesso para AWS CodeCommit](#)

 Note

Se você quiser usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Etapa 2: Instalar o Git

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em. [Solução de problemas](#)

Se a versão do Git que você instalou não incluir um emulador de Bash, como o Git Bash, instale um. Use esse emulador em vez da linha de comando do Windows ao configurar conexões SSH.

Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit

Para configurar as chaves públicas e privadas para o Git e CodeCommit no Windows

1. Abra o emulador de Bash.

Note

Talvez seja necessário executar o emulador com permissões administrativas.

No emulador, execute o comando `ssh-keygen` e siga as instruções para salvar o arquivo no diretório `.ssh` do seu perfil.

Por exemplo:

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

```

Isso gera:

- O `codecommit_rsa` arquivo, que é o arquivo de chave privada.
- O `codecommit_rsa` arquivo.pub, que é o arquivo de chave pública.

Tip

Por padrão, o `ssh-keygen` gera uma chave de 2048 bits. Você pode usar os parâmetros `-t` e `-b` para especificar o tipo e o comprimento da chave. Se você quiser uma chave de 4096 bits no formato `rsa`, especifique isso executando o comando com os seguintes parâmetros:

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

Para obter mais informações sobre os formatos e comprimentos necessários para as chaves SSH, consulte Como [usar o IAM com](#). CodeCommit

2. Execute os comandos a seguir para exibir o valor do arquivo de chave pública (`codecommit_rsa.pub`):

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

Copie o conteúdo do arquivo e, em seguida, feche o Bloco de Notas sem salvar. O conteúdo do arquivo é semelhante ao seguinte:

```
ssh-rsa EXAMPLE-AfICCQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTsBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ2YWN0dGx1MQ8wDQYDVQKQEG5vb25lQGFtYXpvaW5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE user-name@computer-name
```

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

 Note

Você pode visualizar e gerenciar diretamente suas CodeCommit credenciais em Minhas Credenciais de Segurança. Para obter mais informações, consulte [Visualizar e gerenciar suas credenciais](#).

4. No painel de navegação do console do IAM, selecione Usuários e, na lista de usuários, escolha o seu usuário do IAM.
5. Na página de detalhes do usuário, selecione a guia Security Credentials e, em seguida, Upload SSH public key.
6. Cole os conteúdos da chave SSH pública no campo e selecione Upload SSH public key.
7. Copie ou salve as informações no ID da chave SSH (por exemplo, `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 você tiver mais de uma chave SSH IDs carregada, as chaves serão listadas alfabeticamente por ID da chave, não pela data de upload. Verifique se você copiou o ID da chave que está associado à data de upload correta.

8. No emulador de Bash, execute os seguintes comandos para criar um arquivo de configuração no diretório ~/.ssh ou edite-o, caso já exista um:

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

9. Adicione as seguintes linhas ao arquivo, em que o valor para *User* é o ID da chave SSH que você copiou anteriormente e o valor para *IdentityFile* é o caminho e o nome do arquivo de chave privada:

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

 Note

Se você deu ao seu arquivo de chave privada um nome diferente de *codecommit_rsa*, não se esqueça de usá-lo aqui.

Você pode configurar o acesso SSH aos repositórios em várias contas da Amazon Web Services. Para obter mais informações, consulte [Solução de problemas de conexões SSH com AWS CodeCommit](#).

Salve o arquivo como config (não config.txt) e feche o Bloco de Notas.

 Important

O nome do arquivo deve ser `config` sem extensão de arquivo. Caso contrário, as conexões SSH falharão.

10. Execute o seguinte comando para testar a sua configuração de SSH:

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

Você será solicitado a confirmar a conexão porque `git-codecommit.us-east-2.amazonaws.com` ainda não está incluído em seu arquivo de hosts conhecidos. A impressão digital CodeCommit do servidor é exibida como parte da verificação (a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e para MD5 ou 31B1W2g5xn/NA2Ck6dyeJIrQOWvn7n8UES56fG6ZIZQ para SHA256).

 Note

CodeCommit as impressões digitais do servidor são exclusivas para cada Região da AWS. Para ver as impressões digitais do servidor de uma Região da AWS, consulte [Impressões digitais do servidor para CodeCommit](#).

Após confirmar a conexão, aparecerão a confirmação de que você adicionou o servidor ao seu arquivo hosts conhecido e uma mensagem de conexão bem-sucedida. Se você não ver uma mensagem de sucesso, verifique se você salvou o `config` arquivo no diretório `~/.ssh` do usuário do IAM ao qual você configurou para acesso CodeCommit, se o `config` arquivo não tem extensão de arquivo (por exemplo, ele não deve ser chamado de `config.txt`) e se você especificou o arquivo de chave privada correto (`codecommit_rsa` e não `.pub`).

Para solucionar problemas, execute o comando `ssh` com o parâmetro `-v`. Por exemplo:

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

Para obter informações para ajudar a solucionar problemas de conexão, consulte [Solução de problemas de conexões SSH com AWS CodeCommit](#).

Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já enviou para você o nome e os detalhes de conexão do repositório do CodeCommit, ignore esta etapa e clone o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.
 - Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
 - Copie o URL HTTPS (GRC) se estiver usando o comando git-remote-codecommit no computador local.
 - Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta no Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. No emulador Bash, execute o comando git clone com o URL SSH copiado para clonar o repositório. Esse comando cria o repositório local em um subdiretório do diretório onde o comando é executado. Por exemplo, para clonar um repositório com o nome *MyDemoRepo* de um repositório local nomeado *my-demo-repo* na região Leste dos EUA (Ohio):

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

Como alternativa, abra um prompt de comando e, usando a URL e o ID de chave SSH da chave pública que você fez upload no IAM, execute o comando `git clone`. O repositório local é criado em um subdiretório do diretório onde o comando é executado. Por exemplo, para clonar um repositório chamado *MyDemoRepo* para um repositório local chamado *my-demo-repo*:

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

Para obter mais informações, consulte [Conecte-se ao CodeCommit repositório clonando o repositório](#) e [Criar uma confirmação](#).

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Etapas de configuração para conexões HTTPS com AWS CodeCommit repositórios no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI

Antes de se conectar AWS CodeCommit pela primeira vez, você deve concluir as etapas iniciais de configuração. Para a maioria dos usuários, isso pode ser feito mais facilmente seguindo as etapas em [Para usuários de HTTPS usando credenciais do Git](#). No entanto, se você quiser se conectar CodeCommit usando uma conta raiz, acesso federado ou credenciais temporárias, você pode usar o auxiliar de credenciais incluído no AWS CLI

Note

Embora o auxiliar de credenciais seja um método compatível para se conectar CodeCommit usando acesso federado, um provedor de identidade ou credenciais temporárias, o método recomendado é instalar e usar o utilitário `git-remote-codecommit`. Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Tópicos

- [Etapa 1: Configuração inicial para CodeCommit](#)
- [Etapa 2: Instalar o Git](#)
- [Etapa 3: Configurar o assistente de credenciais](#)
- [Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório](#)
- [Próximas etapas](#)

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para configurar uma conta da Amazon Web Services, criar e configurar um usuário do IAM e instalar a AWS CLI.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#).

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. git-remote-codecommit

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```


Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, certifique-se de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)
- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

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

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Etapa 2: Instalar o Git

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações

Importante Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em. [Solução de problemas](#)

Etapa 3: Configurar o assistente de credenciais

1. No terminal, use o Git para executar `git config`, especificando o uso do auxiliar de credenciais do Git com o perfil de credencial e permitindo que o auxiliar de AWS credenciais do Git envie o caminho para os repositórios:

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

Tip

O auxiliar de credencial usa o perfil de AWS credencial padrão ou a função de instância do Amazon EC2. É possível especificar um perfil a ser usado, como `CodeCommitProfile`, se você tiver criado um perfil de credenciais da AWS para ser usado com o CodeCommit:

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

Se o nome do seu perfil contém espaços, não se esqueça de colocá-lo entre aspas ("). É possível configurar perfis por repositório em vez de globalmente, usando `--local`, em vez de `--global`.

O auxiliar de credenciais do Git escreve o seguinte valor para `~/.gitconfig`:

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

```
UseHttpPath = true
```

 Important

Se você quiser usar um usuário do IAM diferente na mesma máquina local CodeCommit, execute o `git config` comando novamente e especifique um perfil de AWS credencial diferente.

2. Execute `git config --global --edit` para verificar se o valor anterior foi escrito para `~/.gitconfig`. Se houver êxito, você verá o valor anterior (além de valores que já possam existir no arquivo de configuração global do Git). Para sair, normalmente, você digita `:q` e pressiona Enter.

Se você tiver problemas depois de configurar seu auxiliar de credencial, consulte [Solução de problemas](#).

 Important

Se você está usando macOS, use as seguintes etapas para garantir que o assistente de credenciais esteja configurado corretamente.

3. Se você estiver usando o macOS, use HTTPS para [se conectar a um CodeCommit](#) repositório. Depois de se conectar a um CodeCommit repositório com HTTPS pela primeira vez, o acesso subsequente falhará após cerca de 15 minutos. A versão padrão do Git no macOS usa o utilitário Acesso às Chaves para armazenar credenciais. Por medidas de segurança, a senha gerada para acessar seu CodeCommit repositório é temporária, portanto, as credenciais armazenadas no chaveiro param de funcionar após cerca de 15 minutos. Para impedir que essas credenciais expiradas sejam usadas, você deve:

- Instalar uma versão do Git que não utilize o conjunto de chaves por padrão; ou
- Configure o utilitário Keychain Access para não fornecer credenciais para CodeCommit repositórios.

1. Abra o utilitário Acesso às Chaves. (Você pode usar o Finder para localizá-lo.)
2. Pesquise por `git-codecommit.us-east-2.amazonaws.com`. Destaque a linha, abra o menu de contexto ou clique com o botão direito do mouse e escolha Obter Informações.
3. Escolha a guia Controle de Acesso.

4. Em Confirmar antes de permitir o acesso, escolha `git-credential-osxkeychain` e, em seguida, selecione o sinal de menos para removê-la da lista.

Note

Após remover `git-credential-osxkeychain` da lista, você verá uma mensagem pop-up sempre que executar um comando do Git. Escolha Negar para continuar. Se você achar os pop-ups um incômodo, aqui estão outras opções:

- Conecte-se CodeCommit usando SSH em vez de HTTPS. Para obter mais informações, consulte [Para conexões SSH no Linux, macOS ou Unix](#).
- No utilitário Acesso às Chaves, na guia Controle de Acesso, para `git-codecommit.us-east-2.amazonaws.com`, escolha a opção Permitir que todos os aplicativos acessem este item (o acesso a esse item não é restrito). Isso impede os pop-ups, mas as credenciais expirarão eventualmente (em média, em cerca de 15 minutos) e você verá uma mensagem de erro 403. Quando isso ocorrer, você deverá excluir o item do conjunto de chaves para restaurar a funcionalidade.
- Para obter mais informações, consulte [Git para macOS: eu configurei o assistente de credenciais com êxito, mas agora o acesso ao meu repositório é negado \(403\)](#).

Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já tiver enviado a você o nome e os detalhes da conexão do CodeCommit repositório, você pode pular essa etapa e clonar o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.
 - Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.

- Copie o URL HTTPS (GRC) se estiver usando o comando `git-remote-codecommit` no computador local.
- Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta na Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. Abra um terminal e execute o comando `git clone` com o URL HTTPS copiado. Por exemplo, para clonar um repositório com o nome *MyDemoRepo* de um repositório local nomeado *my-demo-repo* na região Leste dos EUA (Ohio):

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

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Etapas de configuração para conexões HTTPS com AWS CodeCommit repositórios no Windows com o auxiliar de AWS CLI credenciais

Antes de se conectar AWS CodeCommit pela primeira vez, você deve concluir as etapas iniciais de configuração. Para a maioria dos usuários, isso pode ser feito mais facilmente seguindo as etapas em [Para usuários de HTTPS usando credenciais do Git](#). No entanto, se você quiser se conectar CodeCommit usando uma conta raiz, acesso federado ou credenciais temporárias, você pode usar o auxiliar de credenciais incluído no AWS CLI

 Note

Embora o auxiliar de credenciais seja um método compatível para se conectar CodeCommit usando acesso federado, um provedor de identidade ou credenciais temporárias, o método recomendado é instalar e usar o utilitário. `git-remote-codecommit` Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Este tópico explica as etapas para instalar o AWS CLI, configurar seu computador e AWS perfil, conectar-se a um CodeCommit repositório e clonar esse repositório em seu computador, também conhecido como criação de um repositório local. Se não estiver familiarizado com o Git, revise as informações em [Como posso saber mais sobre o Git?](#).

Tópicos

- [Etapa 1: Configuração inicial para CodeCommit](#)
- [Etapa 2: Instalar o Git](#)
- [Etapa 3: Configurar o assistente de credenciais](#)
- [Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório](#)
- [Próximas etapas](#)

Etapa 1: Configuração inicial para CodeCommit

Siga estas etapas para configurar uma conta da Amazon Web Services, criar e configurar um usuário do IAM e instalar a AWS CLI. AWS CLI Isso inclui um auxiliar de credencial que você configura para conexões HTTPS com seus CodeCommit repositórios.

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#)

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. git-remote-codecommit

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

 Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```

Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, não se esqueça de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

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

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

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)

- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Etapa 2: Instalar o Git

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como [Git for Windows](#). Se você usar esse link para instalar o Git, poderá aceitar todas as configurações padrão da instalação, exceto as seguintes:

- Quando solicitado durante a etapa Ajustar o seu ambiente PATH, escolha a opção de usar o Git na linha de comando.
- (Opcional) Se você pretende usar HTTPS com o auxiliar de credenciais incluído no, em AWS CLI vez de configurar as credenciais CodeCommit do Git, na página Configurando opções extras, verifique se a opção Habilitar o Git Credential Manager está desmarcada. O Gerenciador de Credenciais do Git só é compatível CodeCommit se os usuários do IAM configurarem as credenciais do Git. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#) e [Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado \(403\)](#).

 Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em. [Solução de problemas](#)

Etapa 3: Configurar o assistente de credenciais

AWS CLI Isso inclui um auxiliar de credencial do Git com o qual você pode usar. CodeCommit O auxiliar de credenciais do Git requer um perfil de AWS credencial, que armazena uma cópia do ID da chave de acesso e da chave de AWS acesso AWS secreta de um usuário do IAM (junto com um Região da AWS nome padrão e formato de saída padrão). O auxiliar de credenciais do Git usa essas informações para se autenticar automaticamente, CodeCommit então você não precisa inserir essas informações toda vez que usar o Git para interagir com. CodeCommit

1. Abra um prompt de comando e use o Git para executargit config, especificando o uso do auxiliar de credenciais do Git com o perfil de credencial, o que permite que o auxiliar de AWS credenciais do Git envie o caminho para os repositórios:

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

O assistente de credenciais do Git grava o seguinte no arquivo .gitconfig:

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

 Important

- Se você estiver usando um emulador de Bash em vez da linha de comando do Windows, deve usar aspas simples em vez de aspas duplas.
- O auxiliar de credencial usa o AWS perfil padrão ou a função de instância do Amazon EC2. Se você criou um perfil de AWS credencial para usar,

como *CodeCommitProfile*, você pode modificar o comando da seguinte forma para usá-lo em vez disso:

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

Isso grava o seguinte no arquivo .gitconfig:

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

- Se o nome do seu perfil tiver espaços, será necessário editar o arquivo .gitconfig depois de executar esse comando para inseri-lo em aspas simples ('). Caso contrário, o assistente de credenciais não funcionará.
- Se a instalação do Git para Windows incluir o utilitário Git Credential Manager, você verá erros 403 ou solicitações para fornecer credenciais no utilitário Credential Manager após as primeiras tentativas de conexão. A maneira mais confiável de resolver esse problema é desinstalar e reinstalar o Git para Windows sem a opção do utilitário Git Credential Manager, pois ele não é compatível com o CodeCommit. Se você quiser manter o utilitário Git Credential Manager, você deve executar etapas adicionais de configuração para usá-lo também CodeCommit, incluindo a modificação manual do arquivo .gitconfig para especificar o uso do auxiliar de credenciais ao se conectar a. AWS CodeCommit Exclua todas as credenciais armazenadas do utilitário Credential Manager (o utilitário pode ser encontrado no Painel de Controle). Depois de remover todas as credenciais armazenadas, adicione o seguinte ao seu arquivo .gitconfig, salve-o e tente se conectar novamente a partir de uma nova janela do prompt de comando:

```
[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
```

Além disso, talvez seja necessário redefinir as configurações de git config especificando `--system` em vez de `--global` ou `--local` para que todas as conexões funcionem como esperado.

- Se você quiser usar diferentes usuários do IAM na mesma máquina local CodeCommit, especifique `git config --local` em vez de `git config --global` e execute a configuração para cada perfil de AWS credencial.

2. Execute `git config --global --edit` para verificar se os valores anteriores foram gravados no arquivo `.gitconfig` para seu perfil de usuário (por padrão, `%HOME%\.gitconfig` ou `drive:\Users\UserName\.gitconfig`). Se houver êxito, você verá os valores anterior (além de valores que já possam existir no arquivo de configuração global do Git). Para sair, normalmente, você digita `:q` e pressiona Enter.

Etapa 4: Conectar-se ao CodeCommit console e clonar o repositório

Se um administrador já tiver enviado a você o nome e os detalhes da conexão do CodeCommit repositório, você pode pular essa etapa e clonar o repositório diretamente.

Para se conectar a um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.
 - Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
 - Copie o URL HTTPS (GRC) se estiver usando o comando `git-remote-codecommit` no computador local.
 - Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

 Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta na Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. Abra um prompt de comando e execute o comando `git clone` com a URL HTTPS que você copiou. O repositório local é criado em um subdiretório do diretório onde o comando é executado. Por exemplo, para clonar um repositório com o nome *MyDemoRepo* de um repositório local nomeado *my-demo-repo* na região Leste dos EUA (Ohio):

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

Em algumas versões do Windows, poderá ser exibida uma mensagem pop-up solicitando seu nome de usuário e sua senha. Esse é o sistema de gerenciamento de credenciais interno do Windows, mas ele não é compatível com o assistente de credenciais do AWS CodeCommit. Escolha Cancelar.

Próximas etapas

Você concluiu os pré-requisitos. Siga as etapas [Começando com CodeCommit](#) para começar a usar CodeCommit.

Introdução

A maneira mais fácil de começar CodeCommit é seguir as etapas em [Começando com CodeCommit](#). Se você é novo no Git and CodeCommit, você também deve considerar seguir as etapas em [Começando com o Git e CodeCommit](#). Isso ajuda você a se familiarizar com CodeCommit as noções básicas do uso do Git ao interagir com seus repositórios. CodeCommit

Você também pode acompanhar o tutorial em [Simple Pipeline Walkthrough com CodePipeline e CodeCommit](#) aprender a usar seu CodeCommit repositório como parte de um pipeline de entrega contínua.

Os tutoriais nesta seção foram escritos supondo que você concluiu os [pré-requisitos e a configuração](#), incluindo:

- Atribuir permissões ao usuário do IAM.
- Configuração do gerenciamento de credenciais para conexões HTTPS ou SSH na máquina local que está sendo usada para este tutorial.
- Configurando o AWS CLI se você quiser usar a linha de comando ou o terminal para todas as operações, incluindo a criação do repositório.

Tópicos

- [Começando com AWS CodeCommit](#)
- [Começando com o Git e AWS CodeCommit](#)

Começando com AWS CodeCommit

Este tutorial mostra como usar alguns CodeCommit recursos principais. Primeiro, crie um repositório e confirme algumas alterações nele. Depois disso, procure os arquivos e visualize as alterações. Também é possível criar uma solicitação pull para que outras pessoas possam analisar e comentar as alterações em seu código.

Se você quiser migrar o código existente para CodeCommit, consulte [Migração para o AWS CodeCommit](#).

Se você não está familiarizado com o Git, considere concluir [Começando com o Git e CodeCommit](#) também. Depois de concluir esses tutoriais, você deve ter prática suficiente para começar a usar CodeCommit em seus próprios projetos e em ambientes de equipe.

O CodeCommit console inclui informações úteis em um painel dobrável que você pode abrir a partir do ícone de informações



ou de qualquer link de informações na página. Você pode fechar esse painel a qualquer 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

O CodeCommit console também fornece uma maneira de pesquisar rapidamente seus recursos, como repositórios, criar projetos, aplicativos de implantação e pipelines. Selecione Go to resource (Acessar recurso) ou pressione a tecla / e digite o nome do recurso. Qualquer correspondência aparecerá na lista. As pesquisas não diferenciam letras maiúsculas de minúsculas. Só é possível ver recursos para os quais você tem permissão de visualizar. Para obter mais informações, consulte [Visualizar recursos no console](#).

Pré-requisitos

Antes de começar, conclua os [pré-requisitos e configuração](#), incluindo:

- Atribuir permissões ao usuário do IAM.

- Configuração de gerenciamento de credenciais para conexões HTTPS ou SSH na máquina local que você usar para este tutorial.
- Configurando o AWS CLI se você quiser usar a linha de comando ou o terminal para todas as operações, inclusive para criar o repositório.

Tópicos

- [Etapa 1: criar um CodeCommit repositório](#)
- [Etapa 2: adicionar arquivos ao repositório](#)
- [Etapa 3: navegar pelos conteúdos do seu repositório](#)
- [Etapa 4: criar e colaborar em uma solicitação pull](#)
- [Etapa 5: limpar](#)
- [Etapa 6: próximas etapas](#)

Etapa 1: criar um CodeCommit repositório

Você pode usar o CodeCommit console para criar um CodeCommit repositório. Se você já tem um repositório que deseja usar para este tutorial, ignore esta etapa.

Note

Dependendo do seu uso, você poderá ser cobrado pela criação ou acesso a um repositório. Para obter mais informações, consulte [Preços](#) na página de informações CodeCommit do produto.

Para criar o CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Use o seletor de região para escolher Região da AWS onde você deseja criar o repositório. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), selecione Create repository (Criar repositório).
4. Na página Create repository (Criar repositório), em Repository name (Nome do repositório), insira um nome para o repositório (por exemplo **MyDemoRepo**).

 Note

Os nomes de repositório diferenciam letras maiúsculas de minúsculas e não podem ter mais de 100 caracteres. Para saber mais, consulte [Limites](#).

5. (Opcional) Em Description (Descrição), insira uma descrição, (por exemplo, **My demonstration repository**). Isso pode ajudar você e outros usuários a identificar a finalidade do repositório.
6. (Opcional) Escolha Adicionar tag para adicionar uma ou mais tags de repositório (um rótulo de atributo personalizado que ajuda você a organizar e gerenciar seus AWS recursos) ao seu repositório. Para obter mais informações, consulte [Marcando repositórios em AWS CodeCommit](#).
7. (Opcional) Expanda Configuração adicional para especificar se você deseja usar a chave padrão Chave gerenciada pela AWS ou sua própria chave gerenciada pelo cliente para criptografar e descriptografar os dados nesse repositório. Se você optar por usar sua própria chave gerenciada pelo cliente, deverá garantir que ela esteja disponível no Região da AWS local em que você está criando o repositório e que a chave esteja ativa. Para obter mais informações, consulte [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#).
8. (Opcional) Selecione Ativar Amazon CodeGuru Reviewer para Java e Python se esse repositório contiver código Java ou Python e você quiser que o revisor analise esse código. CodeGuru CodeGuru O revisor usa vários modelos de aprendizado de máquina para encontrar defeitos no código e sugerir automaticamente melhorias e correções nas pull requests. Para obter mais informações, consulte o Guia do usuário do Amazon CodeGuru Reviewer.
9. Escolha Criar.

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 você usar outro nome que não seja MyDemoRepo para o repositório, use-o nas etapas restantes deste tutorial.

Quando o repositório é aberto, você vê informações sobre como adicionar arquivos diretamente do CodeCommit console.

Etapa 2: adicionar arquivos ao repositório

Você pode adicionar arquivos ao repositório da seguinte forma:

- Criação de um arquivo no CodeCommit console. Se você criar o primeiro arquivo para um repositório no console, uma ramificação chamada principal será criada para você. Essa ramificação é a ramificação padrão do seu repositório.
- Fazendo upload de um arquivo do seu computador local usando o CodeCommit console. Se você criar o primeiro arquivo para um repositório no console, uma ramificação chamada principal será criada para você. Essa ramificação é a ramificação padrão do seu repositório.
- Usar um cliente Git para clonar o repositório em seu computador local e, em seguida, adicionar, confirmar e enviar arquivos para o repositório. Uma ramificação será criada para você como parte da primeira confirmação do Git, definida como a ramificação padrão para seu repositório. O nome da ramificação é a escolha padrão do seu cliente Git. Considere configurar seu cliente Git para usar principal como nome da ramificação inicial.

Note

Você pode criar ramificações e alterar a ramificação padrão de um repositório a qualquer momento. Para obter mais informações, consulte [Trabalhando com ramificações em AWS CodeCommit repositórios](#).

A maneira mais simples de começar é abrir o CodeCommit console e adicionar um arquivo. Ao fazer isso, você também cria uma ramificação padrão para seu repositório chamada principal. Para obter instruções sobre como adicionar um arquivo e criar uma primeira confirmação em um repositório usando o AWS CLI, consulte [Criar a primeira confirmação para um repositório usando o AWS CLI](#)

Como adicionar um arquivo a um repositório

1. Na barra de navegação do repositório, escolha Code (Código).
2. Escolha Add file (Adicionar arquivo) e escolha se deseja criar um arquivo ou fazer upload de um arquivo do seu computador. Este tutorial mostra como realizar as duas ações.
3. Para adicionar um arquivo, faça o seguinte:
 - a. Na lista suspensa de ramificações, escolha a ramificação onde deseja adicionar o arquivo. A ramificação padrão é selecionada automaticamente para você. No exemplo mostrado

aqui, a ramificação padrão é nomeada *main*. Se você deseja adicionar o arquivo a outra ramificação, escolha uma ramificação diferente.

- b. Em File name (Nome do arquivo), insira um nome para o arquivo. No editor de código, insira o código do arquivo.
- c. Em Author name (Nome do autor), insira o nome a ser exibido para outros usuários do repositório.
- d. Em Email address (Endereço de e-mail), forneça um endereço de e-mail.
- e. (Opcional) Em Commit message (Mensagem de confirmação), insira uma mensagem breve. Embora seja opcional, recomendamos que você adicione uma mensagem de confirmação para ajudar os membros da equipe a entender porque o arquivo foi adicionado. Se você não inserir uma mensagem de confirmação, uma mensagem padrão será usada.
- f. Escolha Commit changes (Confirmar alterações).

Para fazer upload de um arquivo, faça o seguinte:

- Se você estiver fazendo upload de um arquivo, escolha o arquivo desejado.

The screenshot displays the 'Upload a file' interface in AWS CodeCommit. At the top, there's a header 'MyDemoRepo' with an 'Info' link. Below this is a table with columns 'Name', 'Size', and 'Actions'. The table is currently empty. Below the table, there's a section titled 'Upload file' with the text 'Choose a file to upload.' and a button labeled 'Choose file'. Below this, there's a section titled 'Commit changes to master'. This section contains three input fields: 'Author name' (filled with 'Maria Garcia'), 'Email address' (filled with 'maria_garcia@example.com'), and a 'Commit message - optional' text area (filled with 'Adding my first file to the repository.'). At the bottom right of the interface, there are two buttons: 'Cancel' and 'Commit changes'.

- Em Author name (Nome do autor), insira o nome a ser exibido para outros usuários do repositório.
- Em Email address (Endereço de e-mail), forneça um endereço de e-mail.

- (Opcional) Em Commit message (Mensagem de confirmação), insira uma mensagem breve. Embora seja opcional, recomendamos que você adicione uma mensagem de confirmação para ajudar os membros da equipe a entender porque o arquivo foi adicionado. Se você não inserir uma mensagem de confirmação, uma mensagem padrão será usada.
- Escolha Commit changes (Confirmar alterações).

Para obter mais informações, consulte [Trabalhar com arquivos em repositórios do AWS CodeCommit](#).

Para usar um cliente Git para clonar o repositório, instale o Git no seu computador local e, em seguida, clone o repositório. CodeCommit Adicione alguns arquivos ao repositório e envie-os para o repositório do CodeCommit por push. Para uma introdução detalhada, experimente o [Começando com o Git e CodeCommit](#). Se você estiver familiarizado com o Git, mas não tiver certeza de como fazer isso com um CodeCommit repositório, você pode ver exemplos e instruções em [Criar uma confirmação](#), [Etapa 2: criação de um repositório local](#) ou [Conexão a um repositório](#).

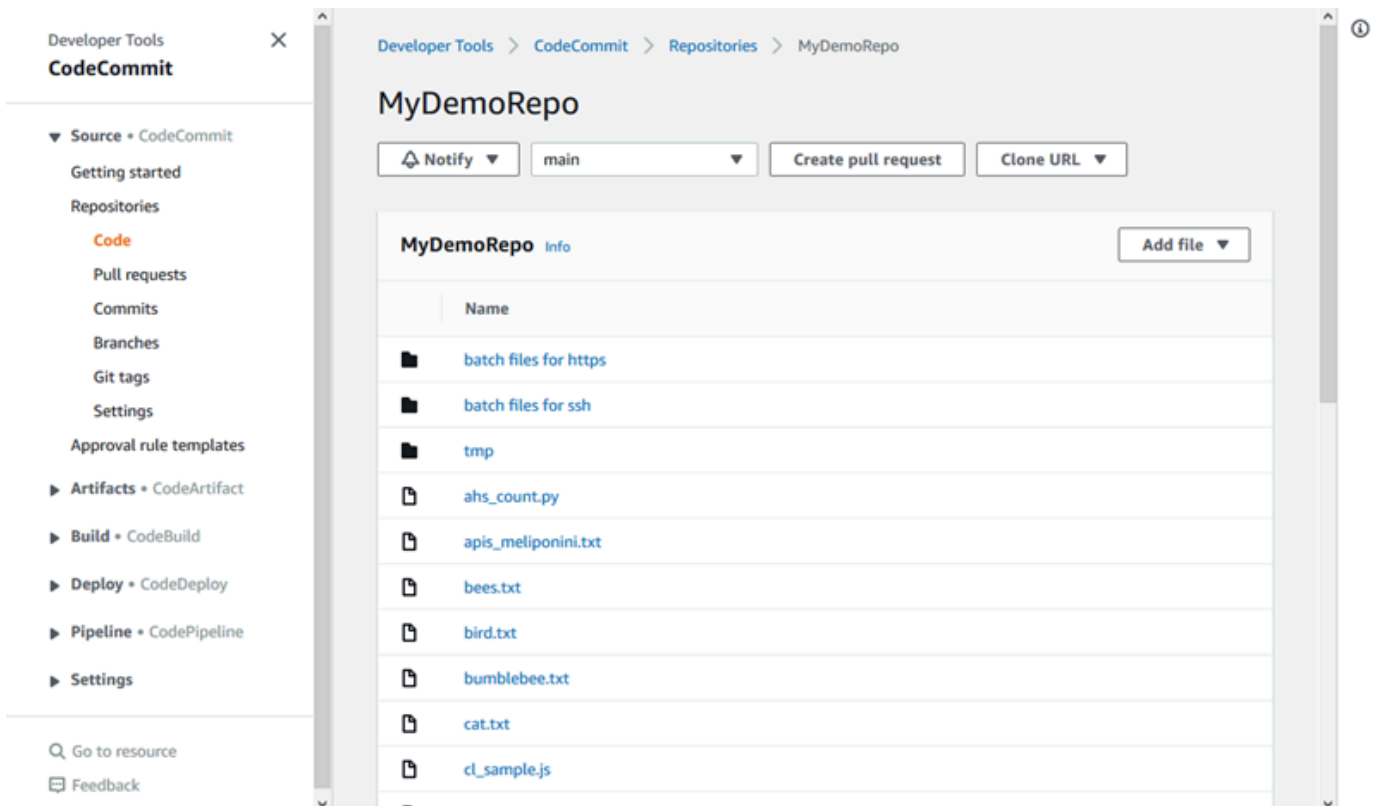
Depois de adicionar alguns arquivos ao CodeCommit repositório, você pode visualizá-los no console.

Etapa 3: navegar pelos conteúdos do seu repositório

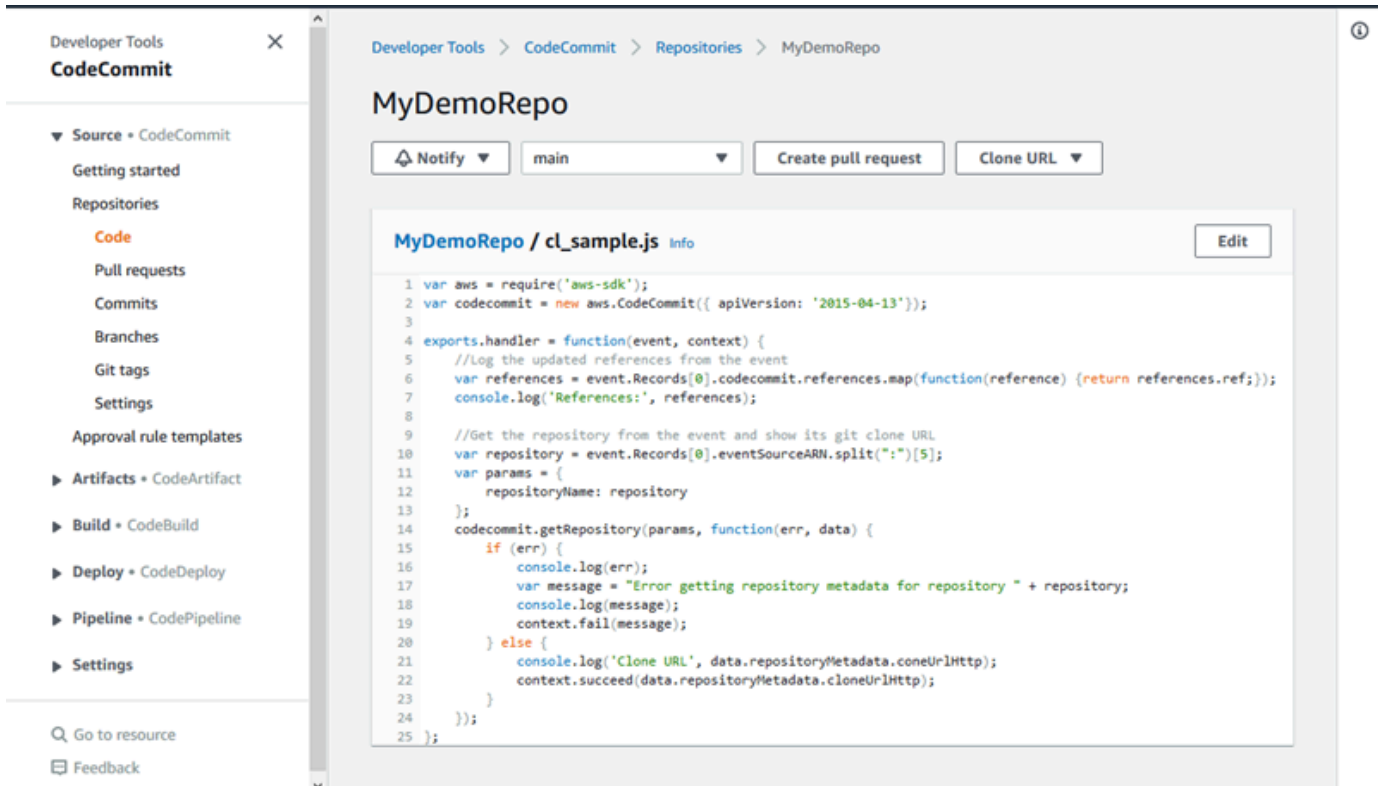
Você pode usar o CodeCommit console para revisar os arquivos em um repositório ou ler rapidamente o conteúdo de um arquivo. Isso ajuda a determinar qual ramificação deve ter o check-out feito ou se você deseja criar uma cópia local de um repositório.

Como navegar no repositório

1. Em Repositórios, escolha MyDemoRepo.
2. A página exibe o conteúdo na ramificação padrão do repositório. Para visualizar outra ramificação ou visualizar o código em uma tag específica, escolha a ramificação ou tag que deseja visualizar na lista. Na captura de tela a seguir, a exibição é definida como a ramificação principal.

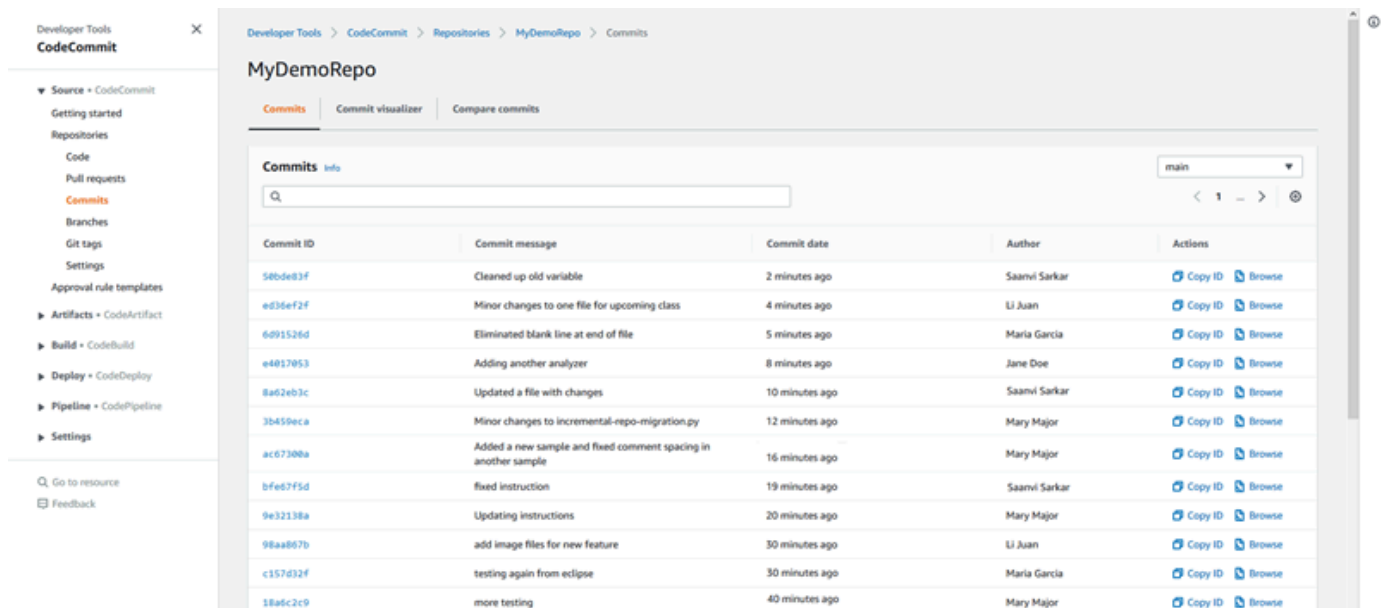


3. Para visualizar os conteúdos de um arquivo no seu repositório, escolha o arquivo na lista. Para alterar a cor do código exibido, escolha o ícone de configurações.



Para obter mais informações, consulte [Procurar arquivos em um repositório](#).

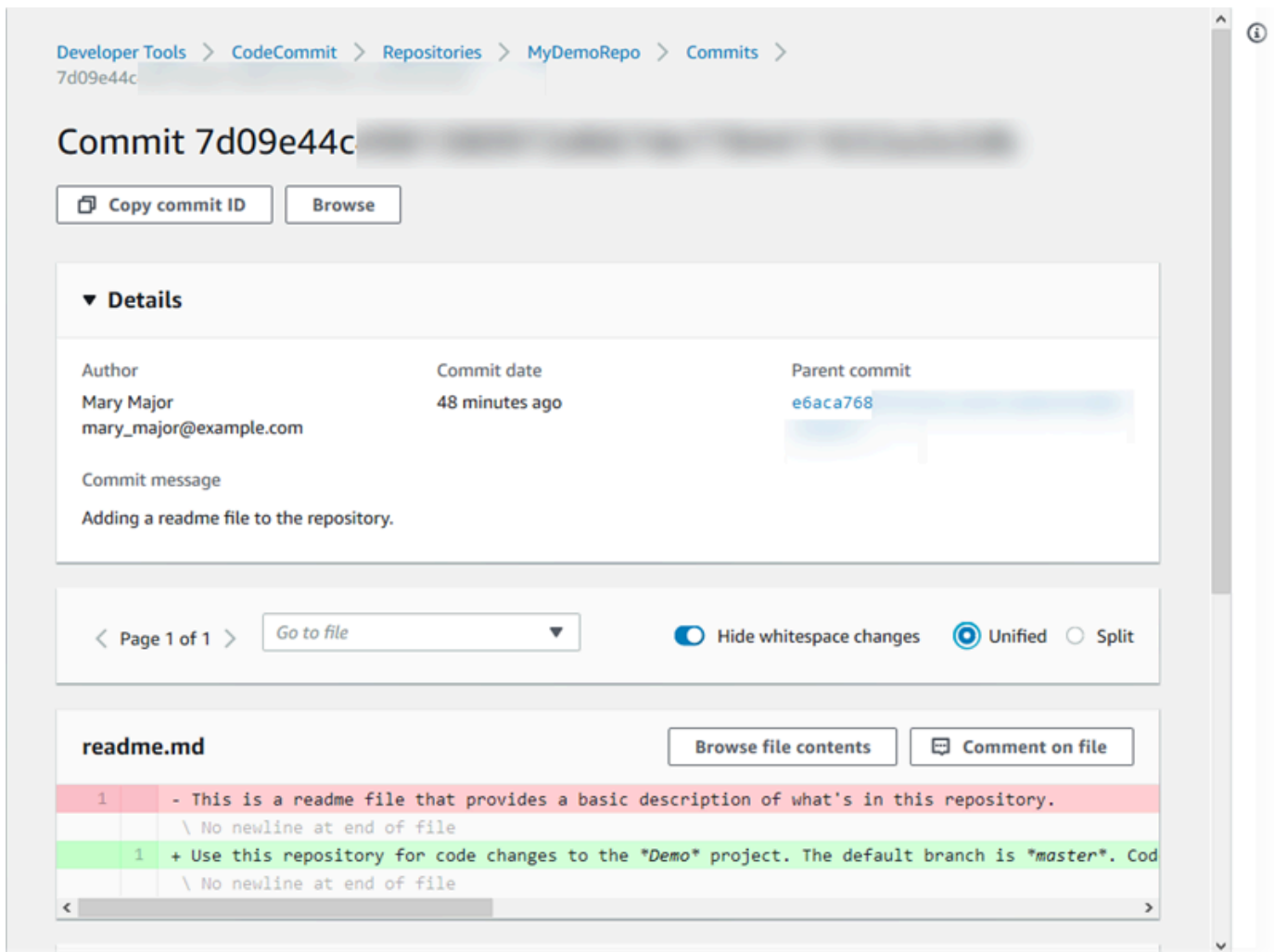
- Para navegar pelo histórico de confirmações do repositório, escolha Commits (Confirmações). O console exibe o histórico de confirmações da ramificação padrão, em ordem cronológica inversa. Revise os detalhes de confirmações por autor, data e muito mais.



- Para visualizar o histórico de confirmações por [ramificação](#) ou [tag do Git](#), escolha a ramificação ou tag que deseja visualizar na lista.
- Para visualizar as diferenças entre uma confirmação e sua confirmação pai, escolha o ID de confirmação abreviado. Você pode escolher como as alterações são exibidas, incluindo mostrar ou ocultar alterações no espaço em branco e se você deseja visualizar alterações embutidas (exibição Unified (Unificada)) ou lado a lado (exibição Split (Dividida)).

Note

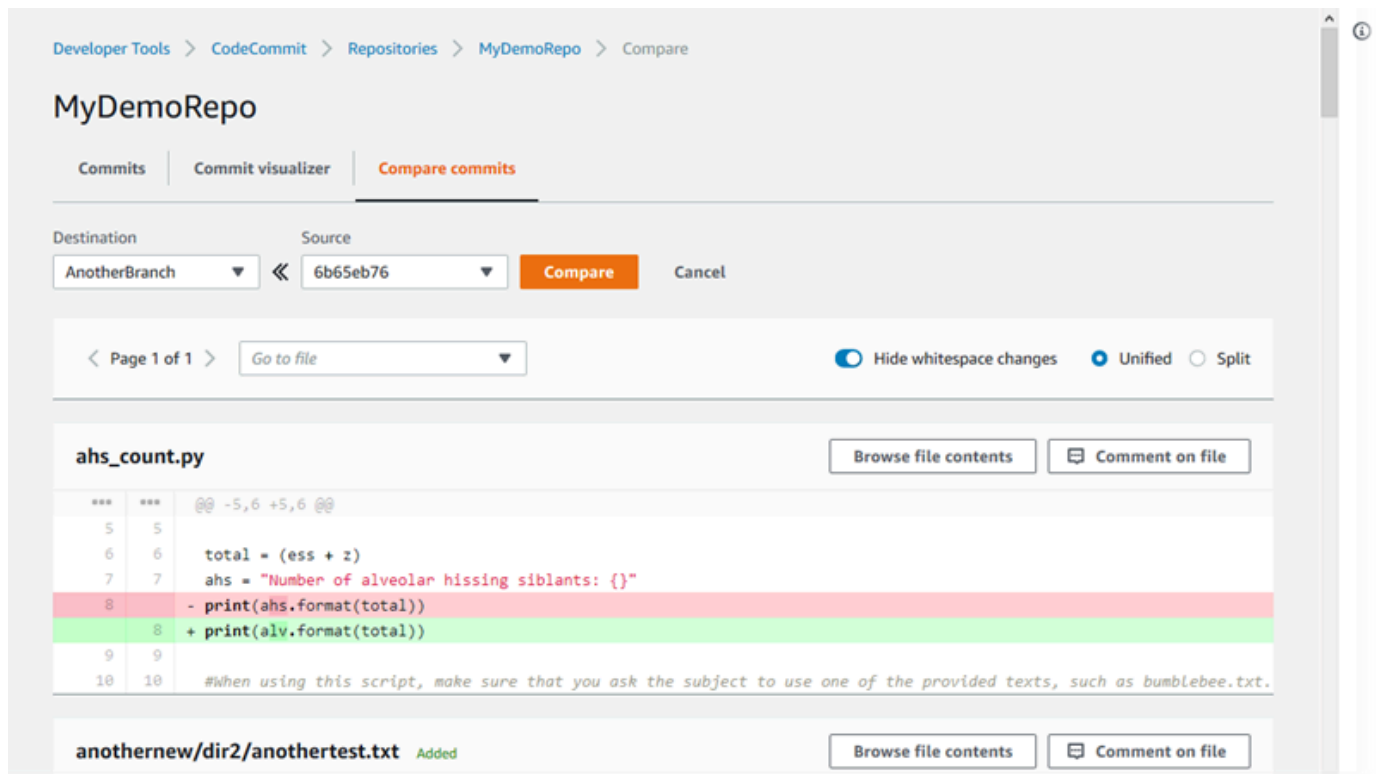
Suas preferências para visualizar código e outras configurações do console são salvas como cookies do navegador sempre que você as altera. Para obter mais informações, consulte [Trabalhar com preferências do usuário](#).



7. Para ver todos os comentários em uma confirmação, escolha a confirmação e, em seguida, role pelas alterações para visualizá-las em linha. Você também pode adicionar seus próprios comentários e responder aos comentários feitos por outros.

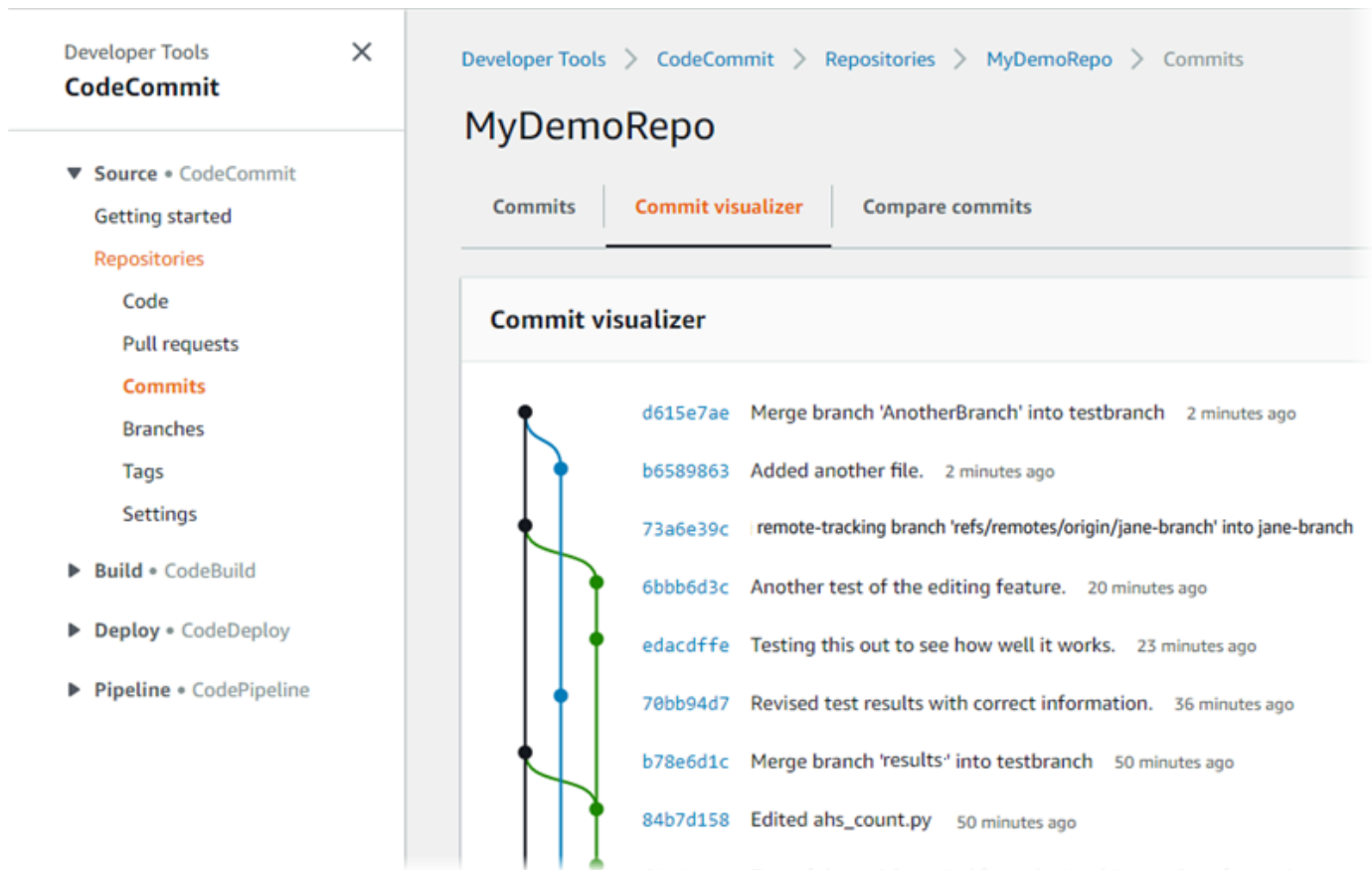
Para obter mais informações, consulte [Comentar sobre uma confirmação](#).

8. Para ver as diferenças entre quaisquer dois especificadores de commits, incluindo tags, branches e commits, no painel de navegação IDs, escolha Commits e, em seguida, escolha Comparar commits.



Para obter mais informações, consulte [Navegar pelo histórico de confirmações de um repositório](#) e [Comparar confirmações](#).

9. Em Commits (Confirmações), escolha a guia Commit visualizer (Visualizador de confirmações).



O gráfico de confirmações será exibido, com a linha de assunto de cada confirmação mostrada próxima ao seu respectivo ponto no gráfico. A exibição da linha de assunto é limitada a 80 caracteres.

10. Para ver mais detalhes sobre uma confirmação, escolha o ID abreviado da confirmação. Para renderizar o gráfico de uma confirmação específica, escolha este ponto no gráfico. Para obter mais informações, consulte [Visualizar um gráfico do histórico de confirmações de um repositório](#).

Etapa 4: criar e colaborar em uma solicitação pull

Ao trabalhar em um repositório com outros usuários, é possível colaborar no código e revisar alterações. É possível criar uma solicitação pull para que outros usuários possam revisar e comentar suas alterações de código em uma ramificação. Também é possível criar uma ou mais regras de aprovação da solicitação pull. Por exemplo, é possível criar uma regra de aprovação que exija que pelo menos dois outros usuários aprovem a solicitação pull para que possa ser mesclada. Depois que a solicitação pull for aprovada, você poderá mesclar essas alterações à ramificação de destino. Se você configurar notificações para seu repositório, os usuários do repositório poderão receber e-mails sobre eventos do repositório (por exemplo, para solicitações pull ou quando alguém fizer

comentários sobre o código). Para obter mais informações, consulte [Configurando notificações para eventos em um repositório AWS CodeCommit](#).

 Important

Antes de criar uma solicitação pull, você deve criar uma ramificação que contenha as alterações de código que deseja rever. Para obter mais informações, consulte [Criar uma ramificação](#).

Como criar e colaborar em uma solicitação pull

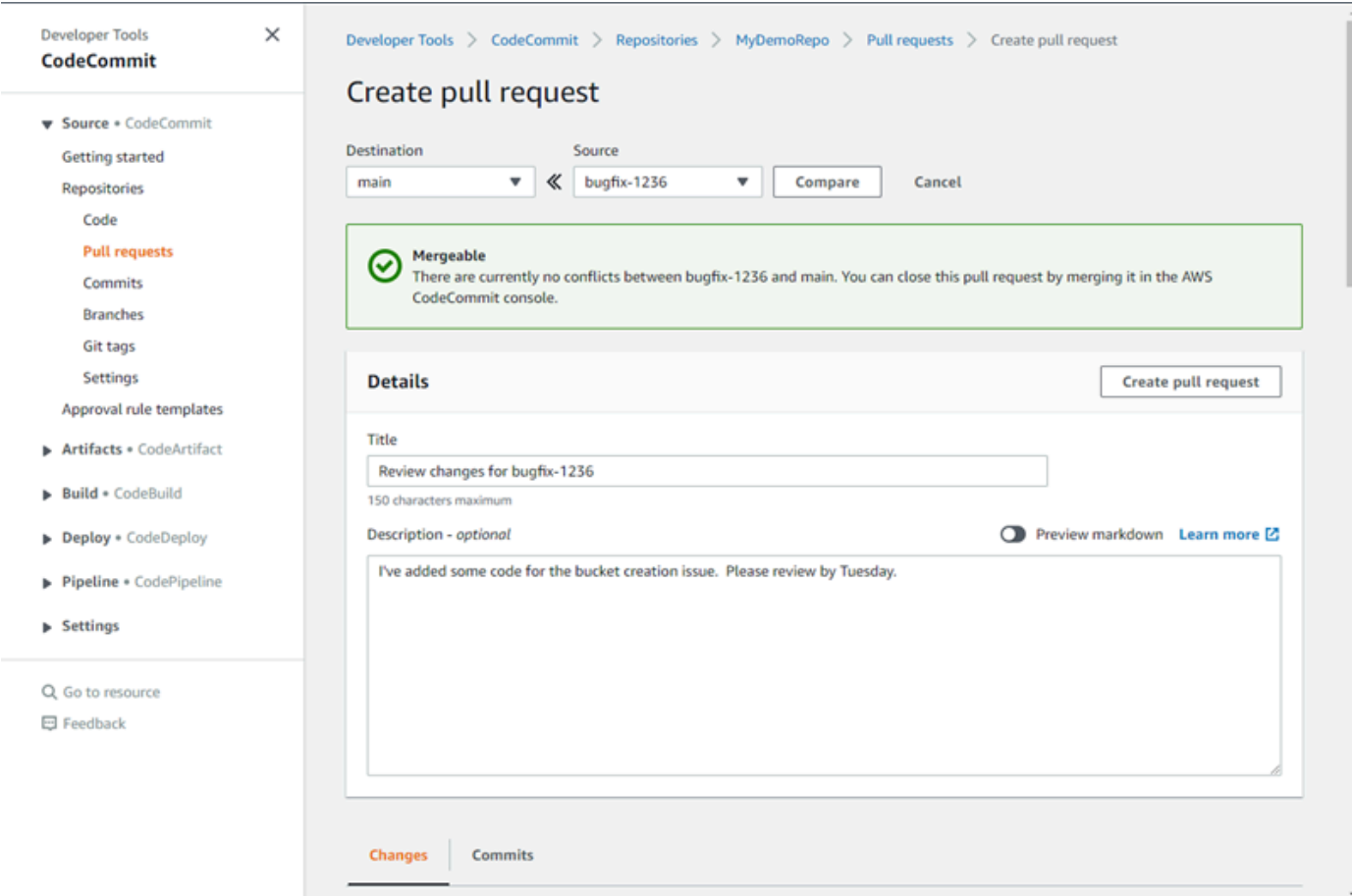
1. No painel de navegação, escolha Pull Requests (Solicitações pull).
2. Em Pull request (Solicitação pull), escolha Create pull request (Criar solicitação pull).

 Tip

Você também pode criar solicitações pull em Branches (Ramificações) e em Code (Código).

Em Create pull request, em Source, escolha a ramificação que contém as alterações que deseja revisar. Em Destination (Destino), escolha a ramificação na qual você deseja que o código seja revisado para ser mesclado quando a solicitação pull for fechada. Escolha Compare.

3. Reveja os detalhes da mesclagem e as alterações para confirmar que a solicitação pull contém as alterações e confirmações que você deseja analisar. Em caso positivo, em Title (Título), insira um título para esta revisão. Este é o título que aparece na lista de solicitações pull para o repositório. Em Description (Descrição), insira detalhes sobre essa revisão e sobre qualquer outra informação útil para os revisores. Escolha Criar.



4. Sua solicitação pull aparece na lista de solicitações pull para o repositório. Você pode filtrar a exibição para mostrar apenas solicitações abertas, solicitações fechadas, solicitações que você criou e muito mais.

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. É possível adicionar uma regra de aprovação à solicitação pull a fim de garantir que determinadas condições sejam atendidas antes que ela seja mesclada. Para adicionar uma regra de aprovação à solicitação pull, escolha a solicitação pull na lista. Na guia Approvals (Aprovações), escolha Create approval rule (Criar regra de aprovação).

6. Em Rule name (Nome da regra), dê à regra um nome descritivo. Por exemplo, se você deseja exigir que duas pessoas aprovem uma solicitação pull para que ela possa ser mesclada, é possível nomear a regra **Require two approvals before merge**. Em Number of approvals needed (Número de aprovações necessárias), insira **2**, o número que você deseja. O padrão é um. Selecione Enviar. Para saber mais sobre regras de aprovação e membros do grupo de aprovações, consulte [Criar uma regra de aprovação de uma solicitação pull](#).

Create approval rule

Rule details

Rule name

Require two approvals before merge

Number of approvals needed

2

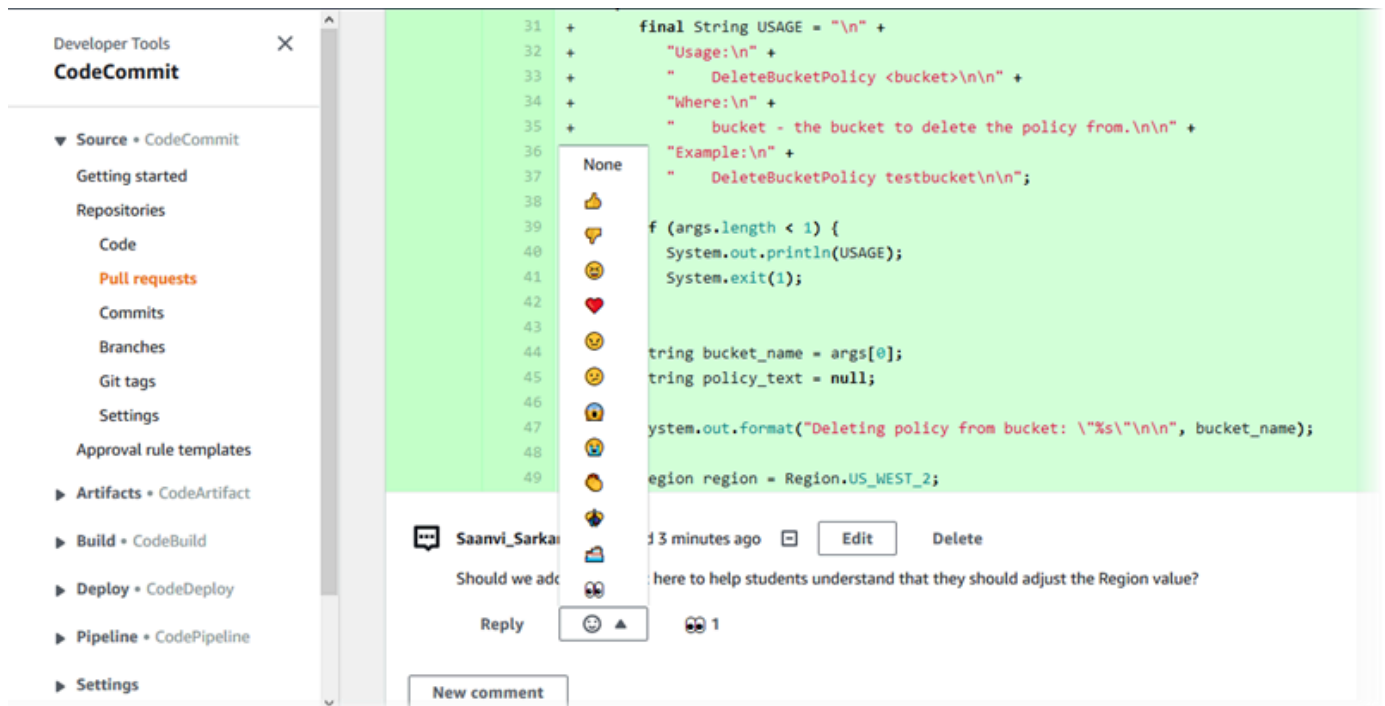
Approval pool members - *optional*

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.

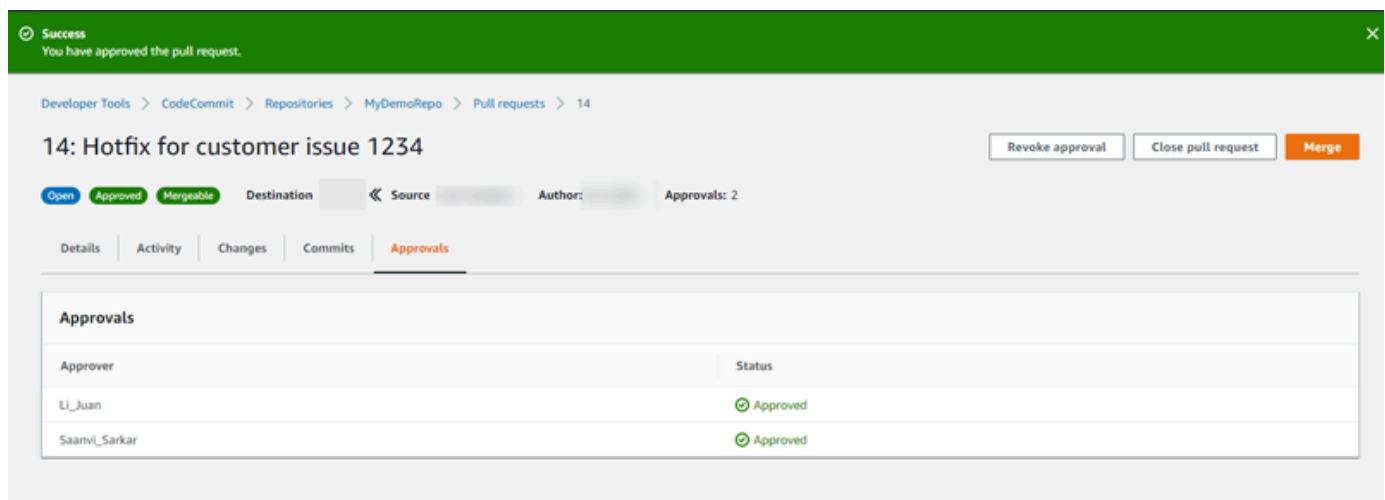
Add

Cancel Submit

7. Se você tiver configurado notificações para o seu repositório e optou por notificar os usuários sobre eventos de solicitação pull, os usuários receberão e-mails sobre sua nova solicitação pull. Os usuários podem visualizar as alterações e comentar em linhas específicas do código, em arquivos e na solicitação pull propriamente dita. Elas também podem responder aos comentários com texto e emojis. Se necessário, você pode inserir as alterações na ramificação da solicitação pull, o que atualiza a solicitação pull.



8. Se você estiver satisfeito com as alterações feitas na solicitação, escolha Approve (Aprovar). É possível optar por aprovar uma solicitação pull mesmo que nenhuma regra de aprovação esteja configurada para ela. Isso fornece um registro claro de que você revisou a solicitação pull e sua aprovação das alterações. Também é possível optar por revogar sua aprovação se mudar de ideia.



Note

Não é possível aprovar uma solicitação pull se você a criou.

9. Quando todas as alterações de código forem analisadas e aceitas apropriadamente, na solicitação pull, siga um destes procedimentos:
 - Se você deseja fechar a solicitação pull sem mesclar ramificações, selecione Close pull request (Fechar solicitação pull).
 - Se deseja mesclar as ramificações e fechar a solicitação pull, escolha Merge (Mesclar).
Será possível escolher entre as estratégias de mesclagem disponíveis para seu código, que dependem das diferenças entre as ramificações de origem e de destino, e a opção de excluir automaticamente a ramificação de origem depois que a mesclagem for concluída. Após ter feito suas escolhas, selecione Merge pull request (Mesclar solicitação pull) para concluir a mesclagem.

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 houver conflitos de mesclagem nas ramificações que não podem ser resolvidos automaticamente, você pode resolvê-los no CodeCommit console ou usar seu cliente Git local para mesclar as ramificações e, em seguida, enviar a mesclagem. Para obter mais informações, consulte [Resolver conflitos em uma solicitação pull em um repositório do AWS CodeCommit](#).

 Note

Você sempre pode mesclar as ramificações manualmente, incluindo ramificações de solicitação pull, usando o comando `git merge` no repositório local e enviando as alterações por push.

Para obter mais informações, consulte [Trabalhar com solicitações pull](#) e [Trabalhar com modelos de regra de aprovação](#).

Etapa 5: limpar

Se você não precisar mais do CodeCommit repositório, exclua o CodeCommit repositório e outros recursos usados neste exercício para não continuar sendo cobrado pelo espaço de armazenamento.

 Important

Esta ação não pode ser desfeita. Após excluir esse repositório, você não poderá mais cloná-lo para nenhum repositório local ou compartilhado. Também não será mais possível extrair dados do repositório ou enviar dados via push para ele, nem executar operações do Git, tanto de repositórios locais como compartilhados.

Se você configurou notificações para o seu repositório, a exclusão do repositório também exclui a regra CloudWatch Amazon Events criada para o repositório. Isso não exclui o tópico do Amazon SNS usado como destino para essa regra.

Se você tiver configurado acionadores para seu repositório, a exclusão do repositório não exclui os tópicos do Amazon SNS nem as funções do Lambda que foram configurados como destinos desses acionadores. Lembre-se de excluir esses recursos se não forem necessários. Para obter mais informações, consulte [Excluir gatilhos de um repositório](#).

Para excluir o CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório que você deseja excluir. Se você tiver seguido a convenção de nomenclatura deste tópico, ele será chamado MyDemoRepo.
3. No painel de navegação, selecione Configurações.

4. Na página Settings, escolha Delete repository em Delete repository.
5. Digite **delete** e, em seguida, escolha Delete (Excluir). O repositório será excluído permanentemente.

Etapa 6: próximas etapas

Agora que você se familiarizou com alguns CodeCommit de seus recursos, considere fazer o seguinte:

- Se você é iniciante no Git CodeCommit e/ou deseja revisar exemplos de uso do Git com CodeCommit, continue com o tutorial. [Começando com o Git e CodeCommit](#)
- Se você quiser trabalhar com outras pessoas em um CodeCommit repositório, consulte [Compartilhar um repositório](#). (Para compartilhar seu repositório com usuários em outra conta da Amazon Web Services, consulte [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#).)
- Se você quiser migrar um repositório para CodeCommit, siga as etapas em. [Migrar para o CodeCommit](#)
- Se você quiser adicionar o repositório a um pipeline de entrega contínua, siga as etapas em [Passo-a-passo de pipeline simples](#).
- Para saber mais sobre produtos e serviços que se integram ao CodeCommit, incluindo exemplos da comunidade, consulte [Integrações de produtos e serviços](#).

Começando com o Git e AWS CodeCommit

Se você é iniciante no Git CodeCommit, este tutorial ajuda você a aprender alguns comandos simples para começar. Se você já está familiarizado com o Git, pode pular este tutorial e ir para [Começando com CodeCommit](#).

Neste tutorial, você cria um repositório que representa uma cópia local do CodeCommit repositório, que chamamos de repositório local.

Depois de criar o repositório local, você fará algumas alterações nele. Em seguida, você envia (envia) suas alterações para o CodeCommit repositório.

Você também simula um ambiente de equipe em que dois usuários confirmam as alterações de forma independente no repositório local e enviam essas alterações para o CodeCommit repositório.

Em seguida, os usuários puxam as alterações do CodeCommit repositório para seu próprio repositório local para ver as alterações feitas pelo outro usuário.

Você também cria ramificações e tags e gerencia algumas permissões de acesso no CodeCommit repositório.

Depois de concluir este tutorial, você terá praticado o suficiente com os principais conceitos do Git e do CodeCommit para usá-los em seus próprios projetos.

Conclua os [pré-requisitos e a configuração](#), incluindo:

- Atribua permissões ao usuário do IAM.
- Configure CodeCommit para se conectar a um repositório usando [HTTPS](#), SSH ou. [git-remote-codecommit](#) Para obter mais informações sobre essas opções, consulte [Configuração do AWS CodeCommit](#).
- Configure o AWS CLI se quiser usar a linha de comando ou o terminal para todas as operações, incluindo a criação do repositório.

Tópicos

- [Etapa 1: criar um CodeCommit repositório](#)
- [Etapa 2: criação de um repositório local](#)
- [Etapa 3: criar sua primeira confirmação](#)
- [Etapa 4: enviar sua primeira confirmação via push](#)
- [Etapa 5: compartilhar o CodeCommit repositório e enviar e receber outro commit](#)
- [Etapa 6: criar e compartilhar uma ramificação](#)
- [Etapa 7: criar e compartilhar uma tag](#)
- [Etapa 8: configurar permissões de acesso](#)
- [Etapa 9: limpar](#)

Etapa 1: criar um CodeCommit repositório

Nesta etapa, você usa o CodeCommit console para criar o repositório.

Você pode pular essa etapa se já tiver um CodeCommit repositório que queira usar.

 Note

Dependendo do seu uso, você poderá ser cobrado pela criação ou acesso a um repositório. Para obter mais informações, consulte [Preços](#) na página de informações CodeCommit do produto.

Para criar o CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Use o seletor de região para escolher Região da AWS onde você deseja criar o repositório. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), selecione Create repository (Criar repositório).
4. Na página Create repository (Criar repositório), em Repository name (Nome do repositório), insira um nome para o repositório (por exemplo **MyDemoRepo**).

 Note

Os nomes de repositório diferenciam letras maiúsculas de minúsculas e não podem ter mais de 100 caracteres. Para saber mais, consulte [Limites](#).

5. (Opcional) Em Description (Descrição), insira uma descrição, (por exemplo, **My demonstration repository**). Isso pode ajudar você e outros usuários a identificar a finalidade do repositório.
6. (Opcional) Escolha Adicionar tag para adicionar uma ou mais tags de repositório (um rótulo de atributo personalizado que ajuda você a organizar e gerenciar seus AWS recursos) ao seu repositório. Para obter mais informações, consulte [Marcando repositórios em AWS CodeCommit](#).
7. (Opcional) Expanda Configuração adicional para especificar se você deseja usar a chave padrão Chave gerenciada pela AWS ou sua própria chave gerenciada pelo cliente para criptografar e descriptografar os dados nesse repositório. Se você optar por usar sua própria chave gerenciada pelo cliente, deverá garantir que ela esteja disponível no Região da AWS local em que você está criando o repositório e que a chave esteja ativa. Para obter mais informações, consulte [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#).
8. (Opcional) Selecione Ativar Amazon CodeGuru Reviewer para Java e Python se esse repositório contiver código Java ou Python e você quiser que o revisor analise esse código. CodeGuru

CodeGuru O revisor usa vários modelos de aprendizado de máquina para encontrar defeitos no código e sugerir automaticamente melhorias e correções nas pull requests. Para obter mais informações, consulte o Guia do usuário do Amazon CodeGuru Reviewer.

9. Escolha Criar.

Note

As etapas restantes deste tutorial são usadas MyDemoRepo para o nome do CodeCommit repositório. Se você escolher um nome diferente, certifique-se de usá-lo durante todo este tutorial.

Para obter mais informações sobre a criação de repositórios, incluindo como criar um repositório do terminal ou da linha de comando, consulte [Criar um repositório](#).

Etapa 2: criação de um repositório local

Nesta etapa, você configurará um repositório local em sua máquina local para se conectar ao seu repositório. Para fazer isso, será preciso selecionar um diretório em sua máquina local que represente o repositório local. Você usa o Git para clonar e inicializar uma cópia do seu CodeCommit repositório vazio dentro desse diretório. Em seguida, você especificará o nome de usuário GIT e endereço de e-mail usados para anotar suas confirmações.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Escolha na lista o repositório ao qual deseja se conectar. Escolha Clone URL (Clonar URL) e escolha o protocolo que você deseja usar ao clonar ou se conectar ao repositório. Isso copia a URL do clone.
 - Copie a URL HTTPS se você estiver usando credenciais do Git com seu usuário do IAM ou o assistente de credenciais incluído na AWS CLI.
 - Copie o URL HTTPS (GRC) se estiver usando o comando git-remote-codecommit no computador local.
 - Copie o URL do SSH se você estiver usando um par de public/private chaves SSH com seu usuário do IAM.

 Note

Se você ver uma página de boas-vindas em vez de uma lista de repositórios, não há repositórios associados à sua AWS conta na Região da AWS local em que você está conectado. Para criar um repositório, consulte [the section called “Criar um repositório”](#) ou siga as etapas no tutorial [Começando com o Git e CodeCommit](#).

4. (Opcional) Recomendamos que você configure seu cliente Git local para usar **main** como nome da ramificação padrão do seu repositório. Esse é o nome usado para a ramificação padrão em todos os exemplos deste guia. Também é o mesmo nome de ramificação padrão CodeCommit usado se você fizer sua primeira confirmação no console. Execute o comando a seguir para configurar o nome da ramificação padrão globalmente para seu sistema:

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

Se você preferir usar um nome de ramificação padrão diferente para todos os seus repositórios, substitua **main** pelo nome de sua preferência. Este tutorial pressupõe que sua ramificação padrão tenha o nome de principal.

Se quiser usar nomes de ramificações padrão diferentes para repositórios diferentes, você pode definir esse atributo localmente (--local) em vez de globalmente (--global).

5. No terminal ou prompt de comando, clone o repositório com o comando `git clone` e forneça a URL do clone copiado na etapa 3. A URL do clone dependerá do protocolo e da configuração usados. Por exemplo, se você estiver usando HTTPS com credenciais do Git para clonar um repositório chamado **MyDemoRepo** na região Leste dos EUA (Ohio):

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

Se você estiver usando HTTPS com `git-remote-codecommit`:

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

Se estiver usando SSH:


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

Note

Se você visualizar um erro ao tentar clonar um repositório, talvez não tenha concluído a configuração necessária no seu computador local. Para obter mais informações, consulte [Configuração do AWS CodeCommit](#).

Etapa 3: criar sua primeira confirmação

Nesta etapa, você criará sua primeira confirmação no repositório local. Para fazer isso, será preciso criar dois arquivos de exemplo no seu repositório local. O Git será usado para preparar e, em seguida, confirmar a alteração no seu repositório local.

1. Use um editor de texto para criar os dois seguintes arquivos de texto de exemplo no seu diretório. Nomeie os arquivos `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. Execute `git config` para adicionar seu nome de usuário e endereço de e-mail representados por espaços reservados `your-user-name` e `your-email-address` ao seu repositório local. Isso facilita a identificação das confirmações realizadas:

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

3. Se você não definiu o nome da ramificação padrão globalmente ao criar o repositório local, execute o comando a seguir para definir o nome da ramificação padrão como **main**:

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

4. Execute `git add` para preparar a alteração:

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

5. Execute `git commit` para confirmar a alteração:

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

 Tip

Para ver os detalhes da confirmação que acabou de fazer, execute `git log`.

Etapa 4: enviar sua primeira confirmação via push

Nesta etapa, você envia o commit do seu repositório local para o seu CodeCommit repositório.

Execute `git push` para enviar seu commit por meio do nome remoto padrão que o Git usa para seu CodeCommit repositório (`origin`), da ramificação padrão em seu repositório local (`:`): `main`

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

 Tip

Depois de enviar os arquivos para o seu CodeCommit repositório, você pode usar o CodeCommit console para visualizar o conteúdo. Para obter mais informações, consulte [Procurar arquivos em um repositório](#).

Etapa 5: compartilhar o CodeCommit repositório e enviar e receber outro commit

Nesta etapa, você compartilha informações sobre o CodeCommit repositório com outro membro da equipe. O integrante da equipe usa essas informações para obter uma cópia local, fazer algumas alterações nela e, em seguida, enviar a cópia local modificada para o seu CodeCommit repositório. Em seguida, você puxa as alterações do CodeCommit repositório para o repositório local.

Neste tutorial, o outro usuário será simulado fazendo com que o Git crie um diretório separado daquele que você criou na [etapa 2](#). (Normalmente, esse diretório está em uma máquina diferente.) Esse novo diretório é uma cópia do seu CodeCommit repositório. Algumas alterações feitas ao diretório existente ou a esse novo diretório são realizadas independentemente. A única maneira de identificar alterações nesses diretórios é extrair do CodeCommit repositório.

Mesmo que estejam na mesma máquina local, chamaremos o diretório existente de repositório local e o novo diretório de repositório compartilhado.

No novo diretório, você obtém uma cópia separada do CodeCommit repositório. Em seguida, você adiciona um novo arquivo de exemplo, confirma as alterações no repositório compartilhado e, em seguida, envia o commit do repositório compartilhado para o seu CodeCommit repositório.

Por fim, você extrairá as alterações do seu repositório para seu repositório local e, em seguida, navegará por ele para ver as alterações confirmadas pelo outro usuário.

1. Mude para o diretório /tmp ou c:\temp.
2. Execute git clone para fazer o pull de uma cópia do repositório para o repositório compartilhado:

Para HTTPS:

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

Para HTTPS com git-remote-codecommit:

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

Para o SSH:

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

Note

Quando um repositório é clonado usando SSH em sistemas operacionais Windows, talvez seja necessário adicionar o ID da chave SSH à cadeia de conexão da seguinte forma:

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

Para obter mais informações, consulte [Para conexões SSH no Windows](#).

Neste comando, MyDemoRepo está o nome do seu CodeCommit repositório. shared-demo-repo é o nome do diretório que o Git cria no /tmp diretório ou no c:\temp diretório. Depois que o Git criar o diretório, o Git extrairá uma cópia do seu repositório no diretório shared-demo-repo.

3. Mude para o diretório shared-demo-repo:

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

4. Execute git config para adicionar outro nome de usuário e endereço de e-mail representado por espaços reservados *other-user-name* e *other-email-address*. Isso facilita a identificação das confirmações realizadas pelo outro usuário:

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

5. Use um editor de texto para criar o seguinte arquivo de texto de exemplo no diretório shared-demo-repo. Nomeie o arquivo horse.txt:

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

6. Execute `git add` para preparar a alteração no repositório compartilhado:

```
git add horse.txt
```

7. Execute `git commit` para confirmar a alteração no repositório compartilhado:

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

8. Execute `git push` para enviar seu commit inicial por meio do nome remoto padrão que o Git usa para seu CodeCommit repositório (`origin`), da ramificação padrão em seu repositório local (`:`):
`main`

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

9. Mude para seu repositório local e execute `git pull` para obter em seu repositório local o commit do repositório compartilhado feito no repositório. CodeCommit Em seguida, execute `git log` para ver a confirmação que foi iniciada a partir do repositório compartilhado.

Etapa 6: criar e compartilhar uma ramificação

Nesta etapa, você cria uma ramificação em seu repositório local, faz algumas alterações e, em seguida, envia a ramificação para seu CodeCommit repositório. Em seguida, você puxa a ramificação do seu CodeCommit repositório para o repositório compartilhado.

Uma ramificação permite que você desenvolva independentemente uma versão diferente do conteúdo do repositório (por exemplo, para trabalhar em um novo recurso de software sem afetar o trabalho dos membros de sua equipe). Quando o recurso fica estável, você mescla a ramificação a outra mais estável do software.

Você usa o Git para criar a ramificação e, em seguida, apontá-la para a primeira confirmação que você fez. Você usa o Git para enviar a ramificação para o CodeCommit repositório. Em seguida, você alternará para o repositório compartilhado e usará o Git para extrair a nova ramificação para o repositório local compartilhado e explorar a ramificação.

1. Do repositório local, execute `git checkout`, especificando o nome da ramificação (por exemplo, `MyNewBranch`) e a ID da primeira confirmação que você fez no repositório local.

Se você não souber o ID da confirmação, execute `git log` para obtê-lo. Certifique-se de que a confirmação tem seu nome de usuário e endereço de e-mail, não o nome de usuário e endereço

de e-mail do outro usuário. Isso é para simular que `main` é uma versão estável do CodeCommit repositório e a `MyNewBranch` ramificação é para algum recurso novo e relativamente instável:

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

2. Execute `git push` para enviar a nova ramificação do repositório local para o CodeCommit repositório:

```
git push origin MyNewBranch
```

3. Agora, efetue o pull da ramificação para o repositório compartilhado e verifique os resultados:
 1. Alterne para o diretório do repositório compartilhado (`shared-demo-repo`).
 2. Faça o pull na nova ramificação (`git fetch origin`).
 3. Confirme se o pull da ramificação foi efetuado (o `git branch --all` exibe uma lista de todas as ramificações do repositório).
 4. Mude para a nova ramificação (`git checkout MyNewBranch`).
 5. Confirme se você mudou para a ramificação `MyNewBranch`, executando `git status` ou `git branch`. O resultado mostra em qual ramificação você está. Neste caso, deve ser `MyNewBranch`.
 6. Visualize a lista de confirmações na ramificação (`git log`).

Aqui está a lista de comandos Git para chamar:

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

4. Alterne de volta para a ramificação `main` e visualize a lista de confirmações. Os comandos do Git devem ter essa aparência:

```
git checkout main  
git log
```

5. Alterne para a ramificação `main` em seu repositório local. Você pode executar `git status` ou `git branch`. O resultado mostra em qual ramificação você está. Neste caso, deve ser `main`. Os comandos do Git devem ter essa aparência:

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

Etapa 7: criar e compartilhar uma tag

Nesta etapa, você cria duas tags em seu repositório local, associa as tags aos commits e, em seguida, envia as tags para o seu CodeCommit repositório. Em seguida, você puxa as alterações do CodeCommit repositório para o repositório compartilhado.

Uma tag é usada para dar um nome legível por humanos a uma confirmação (ou uma ramificação ou até mesmo outra tag). Você poderá fazer isso, por exemplo, se quiser marcar uma confirmação como `v2.1`. Uma confirmação, uma ramificação ou uma tag podem ter várias tags associadas a elas, mas uma tag individual pode ser associada a apenas uma confirmação, uma ramificação ou uma tag. Neste tutorial, você marcará uma confirmação como `release` e outra como `beta`.

Use o Git para criar as tags, apontando a tag `release` para a primeira confirmação que você fez e a tag `beta` para a confirmação feita pelo outro usuário. Em seguida, você usa o Git para enviar as tags para o CodeCommit repositório. Você, então, alternará para o repositório compartilhado e usará o Git para extrair as tags para o repositório local compartilhado e explorar as tags.

1. Do repositório local, execute `git tag`, especificando o nome da nova tag (`release`) e a ID da primeira confirmação que você fez no repositório local.

Se você não souber o ID da confirmação, execute `git log` para obtê-lo. Certifique-se de que a confirmação tem seu nome de usuário e endereço de e-mail, não o nome de usuário e endereço de e-mail do outro usuário. Isso é para simular que seu commit é uma versão estável do CodeCommit repositório:

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

Execute `git tag` novamente para marcar a confirmação do outro usuário com a tag `beta`. Isso é para simular que a confirmação é para algum recurso novo relativamente instável:

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

2. Execute `git push --tags` para enviar as tags para o CodeCommit repositório.
3. Agora, efetue o pull das tags para o repositório compartilhado e verifique os resultados:
 1. Alterne para o diretório do repositório compartilhado (shared-demo-repo).
 2. Faça o pull nas novas tags (`git fetch origin`).
 3. Confirme se foi efetuado o pull das tags (o comando `git tag` exibe uma lista das tags do repositório).
 4. Exiba as informações sobre cada tag (`git log release` e `git log beta`).

Aqui está a lista de comandos Git para chamar:

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

4. Experimente isso também no repositório local:

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

Etapa 8: configurar permissões de acesso

Nesta etapa, você concede permissão ao usuário para sincronizar o repositório compartilhado com o CodeCommit repositório. Esta é uma etapa opcional. É recomendado para usuários interessados em aprender sobre como controlar o acesso aos CodeCommit repositórios quando os usuários usam credenciais do Git ou os pares de chaves SSH são usados com usuários do IAM para acessar os repositórios. CodeCommit

Note

Se você estiver usando acesso federado, credenciais temporárias ou um provedor de identidade da web, como o IAM Identity Center, configure usuários, acessos e permissões para seu provedor de identidade e, em seguida, use `git-remote-codecommit`. Para obter mais

informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#) e [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#).

Para fazer isso, use o console do IAM para criar um usuário que, por padrão, não tem permissões para sincronizar o repositório compartilhado com o CodeCommit repositório. Você pode executar `git pull` para verificar isso. Se o novo usuário não tiver permissão para sincronizar, o comando não funcionará. Em seguida, você voltará ao console do IAM e aplicará uma política que permita ao usuário executar o `git pull`. Você pode executar o `git pull` novamente para verificar isso.

Esta etapa foi escrita supondo que você tenha permissões para criar usuários do IAM em sua conta da Amazon Web Services. Se você não tiver essas permissões, não poderá realizar os procedimentos desta etapa. Avance para [Etapa 9: limpar](#) para limpar os recursos usados no tutorial.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

Certifique-se de fazer login com o mesmo nome de usuário e senha utilizados em [Configurando](#).

2. No painel de navegação, escolha Users e depois Create New Users.
3. Na primeira caixa Enter User Names (Inserir nomes de usuários), insira um nome de usuário de exemplo (por exemplo, **JaneDoe-CodeCommit**). Selecione a caixa Generate an access key for each user (Gerar uma chave de acesso para cada usuário) e selecione Create (Criar).
4. Escolha Show User Security Credentials. Anote o access key ID e a chave de acesso secreta ou escolha Download Credentials.
5. Siga as instruções em [Para usuários de HTTPS usando credenciais do Git](#) para gerar e fornecer as credenciais do usuário do IAM.

Se você quiser usar SSH, siga as instruções em [SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas para Git e CodeCommit](#) ou [Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit](#) para configurar o usuário com chaves públicas e privadas.

6. Executar `git pull`. O seguinte erro deve aparecer:

Para HTTPS:

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

Para o SSH:

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

O erro aparece porque o novo usuário não tem permissão para sincronizar o repositório compartilhado com o CodeCommit repositório.

7. Retorne ao console do IAM. No painel de navegação, escolha Políticas e, em seguida, Create Policy. (Se aparecer um botão Get Started, selecione-o e, em seguida, Create Policy.)
8. Próximo a Create Your Own Policy, escolha Select.
9. Na caixa Policy Name (Nome da política), insira um nome (por exemplo, **CodeCommitAccess-GettingStarted**).
10. Na caixa Documento da política, insira o seguinte para permitir que um usuário do IAM extraia de qualquer repositório associado ao usuário do IAM:

JSON

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

 Tip

Se quiser que o usuário do IAM possa enviar confirmações via push para qualquer repositório associado ao usuário do IAM, insira:

JSON

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

Para obter informações sobre outras permissões de CodeCommit ações e recursos que você pode conceder aos usuários, consulte [Autenticação e controle de acesso para AWS CodeCommit](#).

11. No painel de navegação, escolha Users.
12. Escolha o nome de usuário de exemplo (por exemplo, **JaneDoe-CodeCommit**) ao qual você deseja anexar a política.
13. Escolha a aba Permissões.
14. Em Managed Policies, escolha Attach Policy.
15. Selecione a política **CodeCommitAccess-GettingStarted** que você acabou de criar e escolha Attach Policy (Anexar política).
16. Executar git pull. Dessa vez, o comando deve funcionar e deve aparecer uma mensagem Already up-to-date.
17. Se você estiver usando HTTPS, alterne para suas credenciais originais do Git ou, se estiver usando git-remote-codecommit, seu perfil habitual. Para obter mais informações, consulte as instruções em [Configuração para usuários de HTTPS usando credenciais do Git](#) ou [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Se você estiver usando SSH, alterne para suas chaves originais. Para acessar mais informações, consulte [SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas](#)

[para Git e CodeCommit](#) ou [Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit](#).

Você chegou ao fim deste tutorial.

Etapa 9: limpar

Nesta etapa, você exclui o CodeCommit repositório usado neste tutorial, para não continuar sendo cobrado pelo espaço de armazenamento.

Você também remove o repositório local e o repositório compartilhado em sua máquina local porque eles não serão necessários depois que você excluir o CodeCommit repositório.

Important

Depois de excluir esse repositório, você não poderá cloná-lo para nenhum repositório local ou compartilhado. Também não será possível extrair dados dele nem enviar via push dados para ele de nenhum repositório local ou compartilhado. Esta ação não pode ser desfeita.

Para excluir o CodeCommit repositório (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Na página do Dashboard (Painel), escolha MyDemoRepo na lista de repositórios.
3. No painel de navegação, selecione Configurações.
4. Na página Settings, escolha Delete repository em Delete repository.
5. Na caixa ao lado de Type the name of the repository to confirm deletion (Digite o nome do repositório para confirmar a exclusão), insira **MyDemoRepo** e escolha Delete (Excluir).

Para excluir o CodeCommit repositório (AWS CLI)

Execute o comando [delete-repository](#):

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

Para excluir o repositório local e o repositório compartilhado

Para Linux, macOS ou Unix:

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

Para Windows:

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

Integrações de produtos e serviços ao AWS CodeCommit

Por padrão, o CodeCommit é integrado com diversos serviços da AWS. Você também pode usar o CodeCommit com produtos e serviços fora da AWS. As seguintes informações podem ajudar você a configurar o CodeCommit para integrar com produtos e serviços que você usa.

Tópicos

- [Integração com outros serviços da AWS](#)
- [Exemplos de integração da comunidade](#)

Integração com outros serviços da AWS

O CodeCommit está integrado com os seguintes serviços da AWS:

AWS Amplify

[AWS Amplify](#) O facilita a criação, configuração e implementação de aplicativos móveis escaláveis desenvolvidos pela AWS. O Amplify provisiona e gerencia perfeitamente back-ends móveis e oferece uma estrutura simples para integrar facilmente back-ends a front-ends iOS, Android, web e React Native. Além disso, o Amplify automatiza o processo de lançamento de aplicativos para front-ends e back-ends, o que permite acelerar a entrega de recursos.

Você pode conectar seu repositório do CodeCommit ao console do Amplify. Depois que você autorizar o console do Amplify, o Amplify obterá um token de acesso no provedor do repositório, mas não armazenará o token nos servidores da AWS. O Amplify acessa seu repositório usando chaves de implantação instaladas somente em um repositório específico.

Saiba mais:

- [AWS Amplify Guia do usuário do](#)
- [Conceitos básicos da](#)

AWS Cloud9

[AWS Cloud9](#) O contém um conjunto de ferramentas que você usa para escrever código, compilar, executar, testar, depurar e liberar software na nuvem. Esse conjunto de ferramentas é chamado de ambiente de desenvolvimento integrado AWS Cloud9 ou IDE.

Você tem acesso ao IDE do AWS Cloud9 por meio de um navegador da web. O IDE oferece uma experiência de edição de código completa com suporte para várias linguagens de programação e depuradores de tempo de execução, além de um terminal integrado.

Saiba mais:

- [AWS Cloud9 Guia do usuário do](#)
- [AWS CodeCommit Exemplo do para o AWS Cloud9](#)
- [Integre AWS Cloud9 com AWS CodeCommit](#)

CloudFormation

O [CloudFormation](#) é um serviço que ajuda você a modelar e configurar seus recursos da AWS para despendar menos tempo gerenciando esses recursos e mais tempo se concentrando em seus aplicativos. Você cria um modelo que descreve os recursos, incluindo um repositório do CodeCommit, e o CloudFormation provisiona e configura esses recursos para você.

Saiba mais:

- [AWS CodeCommit Página de recursos do repositório do](#)

AWS CloudTrail

O [CloudTrail](#) captura as chamadas de API da AWS e os eventos relacionados feitos por, ou em nome de, uma conta da Amazon Web Services e entrega arquivos de log a um bucket do Amazon S3 que você especificar. Você pode configurar o CloudTrail para capturar chamadas de API do console do AWS CodeCommit, de comandos do CodeCommit da AWS CLI, do cliente Git local e da API do CodeCommit.

Saiba mais:

- [Registrando chamadas de AWS CodeCommit API com AWS CloudTrail](#)

Amazon CloudWatch Events

O [CloudWatch Events](#) entrega uma transmissão quase em tempo real de eventos do sistema que descrevem as alterações nos recursos da AWS. Com regras simples que você pode configurar rapidamente, é possível corresponder eventos e roteá-los para um ou mais streams ou funções de destino. O CloudWatch Events se torna ciente das alterações operacionais no momento em que ocorrem. O CloudWatch Events responde a essas alterações operacionais e entra em ação conforme necessário, enviando mensagens para responder ao ambiente, ativando funções, fazendo alterações e capturando informações de estado.

Você pode configurar o CloudWatch Events para monitorar repositórios do CodeCommit e responder a eventos do repositório direcionando fluxos, funções, tarefas ou outros processos em outros serviços da AWS, como o Amazon Simple Queue Service, o Amazon Kinesis, o AWS Lambda e muitos outros.

Saiba mais:

- [Guia do usuário do CloudWatch Events](#)
- [AWS CodeCommit Eventos do](#)
- Publicação no blog: [Compilar fluxos de trabalho do AWS CodeCommit com tecnologia a sem servidor usando o Amazon CloudWatch Events e JGit](#)

AWS CodeBuild

O [CodeBuild](#) é um serviço de compilação totalmente gerenciado na nuvem que compila seu código-fonte, executa testes de unidade e produz artefatos prontos para implantação. Você pode armazenar o código-fonte a ser compilado e a especificação de compilação em um repositório do CodeCommit. Você pode usar o CodeBuild diretamente com o CodeCommit, ou pode incorporar o CodeBuild e o CodeCommit a um pipeline de entrega contínua com o CodePipeline.

Saiba mais:

- [Planejar uma compilação](#)
- [Criar um projeto de build](#)
- [Usar o CodePipeline com o AWS CodeBuild para executar compilações](#)

Amazon CodeGuru Reviewer

O Amazon CodeGuru Reviewer é um serviço automatizado de revisão de código que usa análise de programas e machine learning para detectar problemas comuns e recomendar correções em seu código Java ou Python. Você pode associar repositórios na sua conta da Amazon Web Services com o CodeGuru Reviewer. Ao fazer isso, o CodeGuru Reviewer cria uma função vinculada ao serviço que permite que o CodeGuru Reviewer analise códigos em todas as solicitações pull criadas depois que a associação é feita.

Saiba mais:

- [Associar ou desassociar um AWS CodeCommit repositório com o Amazon Reviewer CodeGuru](#)
- [Guia do usuário do Amazon CodeGuru Reviewer](#)

AWS CodePipeline

O [CodePipeline](#) é um serviço de entrega contínua que pode ser usado para modelar, visualizar e automatizar as etapas necessárias para lançar seu software. Você pode configurar o CodePipeline para usar um repositório do CodeCommit como uma ação de origem em um pipeline e automatizar a compilação, teste e implantação das suas alterações.

Saiba mais:

- [Passo a passo do pipeline simples com CodePipeline e AWS CodeCommit](#)
- [Migrar para detecção de alterações do Amazon CloudWatch Events para pipelines com um repositório do CodeCommit](#)
- [Métodos de detecção de alterações usados para iniciar pipelines automaticamente](#)

AWS Elastic Beanstalk

O [Elastic Beanstalk](#) é um serviço gerenciado que facilita a implantação e o gerenciamento de aplicativos na nuvem da AWS sem se preocupar com a infraestrutura que executa esses aplicativos. Você pode usar a interface de linha de comando do Elastic Beanstalk (EB CLI) para implantar seu aplicativo diretamente de um repositório do CodeCommit novo ou existente.

Saiba mais:

- [Usar o EB CLI com o AWS CodeCommit](#)
- [Uso de um repositório do AWS CodeCommit existente](#)
- [eb codesource \(comando do EB CLI\)](#)

AWS Key Management Service

[AWS KMS](#) O é um serviço gerenciado que facilita a criação e o controle de chaves de criptografia usadas para criptografar seus dados. Por padrão, o CodeCommit usa o AWS KMS para criptografar repositórios.

Saiba mais:

- [AWS KMS e criptografia](#)

AWS Lambda

O [Lambda](#) permite executar códigos sem provisionar ou gerenciar servidores. Você pode configurar acionadores para repositórios do CodeCommit que invocam funções do Lambda em resposta a eventos do repositório.

Saiba mais:

- [Criar um gatilho para uma função do Lambda](#)
- [AWS Lambda Guia do desenvolvedor do](#)

Amazon Simple Notification Service

O [Amazon SNS](#) é um serviço web que permite que aplicativos, usuários finais e dispositivos enviem e recebam notificações da nuvem instantaneamente. Você pode configurar os acionadores para repositórios do CodeCommit que enviam notificações do Amazon SNS em resposta a eventos do repositório. Você também pode usar notificações do Amazon SNS para integrar com outros serviços da AWS. Por exemplo, você pode usar uma notificação do Amazon SNS para enviar mensagens para uma fila do Amazon Simple Queue Service.

Saiba mais:

- [Criar um gatilho para um tópico do Amazon SNS](#)
- [Guia do desenvolvedor do Amazon Simple Notification Service](#)

Exemplos de integração da comunidade

As seguintes seções fornecem links para publicações no blog, artigos e exemplos fornecidos pela comunidade.

Note

Esses links são fornecidos apenas para fins informativos e não devem ser considerados como uma lista abrangente ou como um endosso do conteúdo dos exemplos. A AWS não se responsabiliza pelo conteúdo ou pela precisão de conteúdo externo.

Tópicos

- [Publicações no blog](#)
- [Exemplos de código](#)

Publicações no blog

- [Integrar o SonarQube como aprovador de solicitações pull no AWS CodeCommit](#)

Saiba como criar um repositório do CodeCommit que exija uma análise de qualidade bem-sucedida do SonarQube antes que as solicitações pull possam ser mescladas.

Publicado em dezembro de 12, 2019

- [Migrar para o AWS CodeCommit, AWS CodePipeline e AWS CodeBuild do GitLab](#)

Saiba como migrar vários repositórios para o AWS CodeCommit do GitLab e configurar um pipeline CI/CD usando o AWS CodePipeline e o AWS CodeBuild.

Publicado em novembro de 22, 2019

- [Implementação do GitFlow usando o AWS CodePipeline, o AWS CodeCommit, o AWS CodeBuild e o AWS CodeDeploy](#)

Saiba como implementar o GitFlow usando o AWS CodePipeline, AWS CodeCommit, o AWS CodeBuild e o AWS CodeDeploy.

Publicado em fevereiro de 22, 2019

- [Usar o Git com o AWS CodeCommit em várias contas da AWS](#)

Saiba como gerenciar sua configuração do Git em várias contas da Amazon Web Services.

Publicado em fevereiro de 12, 2019

- [Validação de solicitações pull do AWS CodeCommit com o AWS CodeBuild e o AWS Lambda](#)

Saiba como validar solicitações pull com o AWS CodeCommit, o AWS CodeBuild e o AWS Lambda. Ao executar testes em relação às alterações propostas antes de mesclá-las à ramificação padrão, você pode ajudar a garantir um alto nível de qualidade nas solicitações pull, detectar possíveis problemas e aumentar a confiança do desenvolvedor quanto às alterações.

Publicado em fevereiro de 11, 2019

- [Usar identidades federadas com o AWS CodeCommit](#)

Saiba como acessar repositórios do AWS CodeCommit usando as identidades usadas em sua empresa.

Publicado em outubro de 5, 2018

- [Refinar o acesso às ramificações no AWS CodeCommit](#)

Saiba como restringir confirmações em ramificações do repositório criando e aplicando uma política do IAM que usa uma chave de contexto.

Publicado em 16 de maio de 2018

- [Replicar os repositórios do AWS CodeCommit entre regiões usando o AWS Fargate](#)

Saiba como configurar a replicação contínua de um repositório do CodeCommit de uma região da AWS para outra usando uma arquitetura de tecnologia sem servidor.

Publicado em 11 abril de 2018

- [Distribuição da infraestrutura do AWS OpsWorks for Chef Automate](#)

Saiba como usar o CodePipeline, CodeCommit, CodeBuild e AWS Lambda para garantir que livros de receitas e outras configurações sejam implantadas de forma consistente em dois ou mais Chef Servers residindo em uma ou mais Regiões da AWS.

Publicado em 9 de março de 2018

- [Creme de amendoim e chocolate: pipeline de CI/CD de funções do Azure com o AWS CodeCommit](#)

Saiba como criar um pipeline de CI/CD do Azure Functions baseado no PowerShell em que o código é armazenado em um repositório do CodeCommit.

Publicado em 19 de fevereiro de 2018

- [Implantação contínua no Kubernetes usando o AWS CodePipeline, o AWS CodeCommit, o AWS CodeBuild, o Amazon ECR e o AWS Lambda](#)

Saiba como usar o Kubernetes e a AWS de forma conjunta para criar um pipeline de implantação contínua totalmente gerenciado para aplicativos baseados em contêineres.

Publicado em janeiro 11 de 2018

- [Usar solicitações de extração do AWS CodeCommit para solicitar revisões de código e discutir código](#)

Saiba como usar solicitações pull para revisar, comentar e iterar de maneira interativa em alterações de código em um repositório do CodeCommit.

Publicado em 20 novembro de 2017

- [Publicação no blog: Compilar fluxos de trabalho do AWS CodeCommit de tecnologia sem servidor usando o Amazon CloudWatch Events e JGit](#)

Saiba como criar regras do CloudWatch Events que processam alterações em um repositório usando eventos do repositório do CodeCommit e ações de destino em outros serviços da AWS. Os exemplos incluem funções do AWS Lambda que aplicam políticas de mensagens de confirmação do Git em confirmações, replicam um repositório do CodeCommit e fazem backup de um repositório do CodeCommit no Amazon S3.

Publicado em 3 de agosto de 2017

- [Migrar para o AWS CodeCommit](#)

Saiba como enviar códigos via push para dois repositórios como parte da migração de outro repositório do Git para o CodeCommit usando o SourceTree.

Publicado em 6 de setembro de 2016

- [Configuração de teste contínuo com o Appium, o AWS CodeCommit, o Jenkins e o AWS Device Farm](#)

Saiba como criar um processo de teste contínuo para dispositivos móveis usando o Appium, o CodeCommit, o Jenkins e o Device Farm.

Publicado em 2 de fevereiro de 2016

- [Usar o AWS CodeCommit com repositórios do Git em várias contas da Amazon Web Services](#)

Saiba como clonar o repositório do CodeCommit e, com um comando, configurar o assistente de credenciais para usar um perfil do IAM específico para conexões com esse repositório.

Publicado em novembro de 2015

- [Integração do OpsWorks e do AWS CodeCommit](#)

Saiba como o OpsWorks pode buscar automaticamente aplicativos e livros de receitas do Chef no CodeCommit.

Publicado em 25 de agosto de 2015

- [Uso do AWS CodeCommit e dos assistentes de credenciais do GitHub](#)

Saiba como configurar seu arquivo gitconfig para trabalhar com assistentes de credenciais do CodeCommit e do GitHub.

Publicado em setembro de 2015

- [Uso do AWS CodeCommit no Eclipse](#)

Saiba como usar as ferramentas de EGit no Eclipse para trabalhar com o CodeCommit.

Publicado em agosto de 2015

- [AWS CodeCommit com credenciais de função do Amazon EC](#)

Saiba como usar um perfil de instância do Amazon EC2 ao configurar o acesso automatizado de atendentes a um repositório do CodeCommit.

Publicado em julho de 2015

- [Integração do AWS CodeCommit com o Jenkins](#)

Saiba como usar o CodeCommit e o Jenkins para dar suporte a dois cenários de integração contínua (CI) simples.

Publicado em julho de 2015

- [Integração do AWS CodeCommit com o Review Board](#)

Saiba como integrar o CodeCommit em um fluxo de trabalho de desenvolvimento usando o sistema de revisão de código [Review Board](#).

Publicado em julho de 2015

Exemplos de código

Veja a seguir exemplos de código que podem ser úteis para usuários do CodeCommit.

- [Script do Mac OS X para excluir periodicamente credenciais armazenadas em cache na loja de certificados do OS X](#)

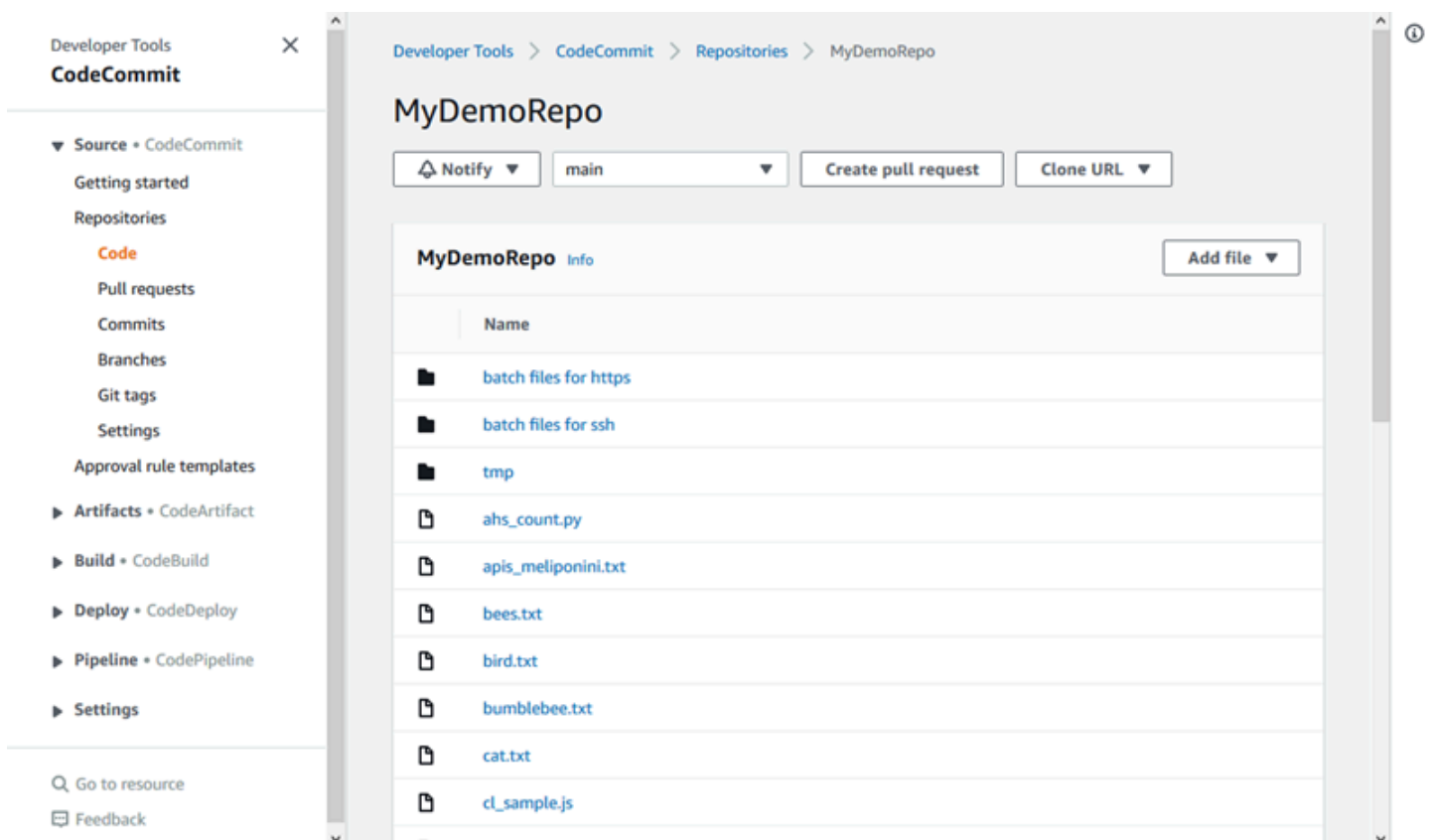
Se você usa o assistente de credenciais do CodeCommit no Mac OS X, provavelmente já está familiarizado com credenciais em cache. Este script demonstra uma solução.

Autor: Nico Coetzee

Publicado em fevereiro de 2016

Trabalhando com repositórios em AWS CodeCommit

Um repositório é o objeto de controle de versão fundamental no CodeCommit. É onde você armazena com segurança códigos e arquivos do seu projeto. Ele também armazena o histórico do projeto, desde a primeira confirmação até as alterações mais recentes. É possível compartilhar seu repositório com outros usuários para trabalharem com você em um projeto. Se você adicionar AWS tags aos repositórios, poderá configurar notificações para que os usuários do repositório recebam e-mails sobre eventos (por exemplo, outro usuário comentando o código). Você também pode alterar as configurações padrão do repositório, consultar seus conteúdos e mais. É possível criar triggers para o repositório de forma que pushes de código ou outros eventos acionem ações, como e-mails ou funções de código. Você pode, inclusive, configurar um repositório no seu computador local (um repositório local) para enviar suas alterações por push para mais de um repositório.



Antes de enviar alterações para um CodeCommit repositório, você deve configurar um usuário do IAM em sua conta da Amazon Web Services ou configurar o acesso para acesso federado ou credenciais temporárias. Para obter mais informações, consulte [Etapa 1: Configuração inicial para CodeCommit](#) e [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Para obter informações sobre como trabalhar com outros aspectos do seu repositório em CodeCommit [Trabalhar com arquivos](#), consulte [Trabalhar com solicitações pull](#), [Trabalhar com confirmações](#), [Como trabalhar com ramificações](#), e [Trabalhar com preferências do usuário](#). Para obter informações sobre a migração para CodeCommit, consulte [Migrar para o CodeCommit](#).

Tópicos

- [Crie um AWS CodeCommit repositório](#)
- [Conecte-se a um AWS CodeCommit repositório](#)
- [Compartilhar um AWS CodeCommit repositório](#)
- [Configurando notificações para eventos em um repositório AWS CodeCommit](#)
- [Marcando repositórios em AWS CodeCommit](#)
- [Gerenciar gatilhos para um repositório AWS CodeCommit](#)
- [Associar ou desassociar um AWS CodeCommit repositório com o Amazon Reviewer CodeGuru](#)
- [Exibir detalhes CodeCommit do repositório](#)
- [Alterar as AWS CodeCommit configurações do repositório](#)
- [Sincronizar alterações entre um repositório local e um repositório AWS CodeCommit](#)
- [Confirmações de push em um repositório do Git adicional](#)
- [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#)
- [Excluir um AWS CodeCommit repositório](#)

Crie um AWS CodeCommit repositório

Use o AWS CodeCommit console ou o AWS Command Line Interface (AWS CLI) para criar um CodeCommit repositório vazio. Para adicionar tags a um repositório depois de criá-lo, consulte [Adicionar uma tag a um repositório](#).

Estas instruções supõem que você tenha concluído as etapas em [Configurando](#).

Note

Dependendo do seu uso, você poderá ser cobrado pela criação ou acesso a um repositório. Para obter mais informações, consulte [Preços](#) na página de informações CodeCommit do produto.

Tópicos

- [Criar um repositório \(console\)](#)
- [Criar um repositório \(AWS CLI\)](#)

Criar um repositório (console)

Para criar um CodeCommit repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde você deseja criar o repositório. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), selecione Create repository (Criar repositório).
4. Na página Create repository (Criar repositório), em Repository name (Nome do repositório), insira um nome para o repositório.

Note

Os nomes de repositórios diferenciam maiúsculas de minúsculas. O nome deve ser exclusivo na Região da AWS para sua conta da Amazon Web Services.

5. (Opcional) Em Description (Descrição), insira uma descrição para o repositório. Isso pode ajudar você e outros usuários a identificar a finalidade do repositório.

Note

O campo de descrição exibe o Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o GetRepository ou BatchGetRepositories APIs e planeja exibir o campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

6. (Opcional) Escolha Adicionar tag para adicionar uma ou mais tags de repositório (um rótulo de atributo personalizado que ajuda você a organizar e gerenciar seus AWS recursos) ao seu repositório. Para obter mais informações, consulte [Marcando repositórios em AWS CodeCommit](#).

7. (Opcional) Expanda Configuração adicional para especificar se você deseja usar a chave padrão Chave gerenciada pela AWS ou sua própria chave gerenciada pelo cliente para criptografar e descriptografar dados nesse repositório. Se você optar por usar sua própria chave gerenciada pelo cliente, deverá garantir que ela esteja disponível no Região da AWS local em que você está criando o repositório e que a chave esteja ativa. Para obter mais informações, consulte [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#).
8. (Opcional) Selecione Habilitar Amazon CodeGuru Reviewer para Java e Python se esse repositório contiver código Java ou Python e você CodeGuru quiser que o Reviewer o analise. CodeGuru O revisor usa vários modelos de aprendizado de máquina para encontrar defeitos no código e sugerir melhorias e correções nas pull requests. Para obter mais informações, consulte o [Guia do usuário do Amazon CodeGuru Reviewer](#).
9. Escolha Criar.

Depois de criar um repositório, você pode se conectar a ele e começar a adicionar código por meio do CodeCommit console ou de um cliente Git local, ou integrando CodeCommit seu repositório ao seu IDE favorito. Para obter mais informações, consulte [Configuração do AWS CodeCommit](#). Você também pode adicionar seu repositório a um pipeline de entrega contínua. Para obter mais informações, consulte [Passo-a-passo de pipeline simples](#).

Para obter informações sobre o novo CodeCommit repositório, como as que serão usadas URLs ao clonar o repositório, escolha o nome do repositório na lista ou simplesmente escolha o protocolo de conexão que você deseja usar ao lado do nome do repositório.

Para compartilhar o repositório com outros, envie o link HTTPS ou SSH a ser usado para clonar o repositório. Verifique se eles têm as permissões necessárias para acessar o repositório. Para obter mais informações, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#).

Criar um repositório (AWS CLI)

Você pode usar o AWS CLI para criar um CodeCommit repositório. Ao contrário do console, você pode adicionar tags a um repositório se criá-lo usando a AWS CLI.

1. Certifique-se de ter configurado o AWS CLI com o Região da AWS local onde o repositório existe. Para verificar a região, execute o seguinte comando na linha de comando ou no terminal e revise as informações referentes ao nome da região padrão:

```
aws configure
```

O nome da região padrão deve corresponder Região da AWS ao do repositório em CodeCommit. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

2. Execute o comando `create-repository`, especificando:

- Um nome que identifica de forma exclusiva o CodeCommit repositório (com a opção). `--repository-name`

 Note

Esse nome precisa ser exclusivo em toda a conta da Amazon Web Services.

- Um comentário opcional sobre o CodeCommit repositório (com a `--repository-description` opção).
- Um par ou pares opcionais de valores-chave para usar como tags para o CodeCommit repositório (com a `--tags` opção).
- Uma chave gerenciada pelo cliente opcional para usar ao criptografar e descriptografar esse repositório. Todos os repositórios são criptografados em trânsito e em repouso usando uma chave no AWS KMS. Se nenhuma chave for especificada, a chave AWS gerenciada padrão `aws/codecommit` será usada.

Por exemplo, para criar um CodeCommit repositório nomeado `MyDemoRepo` com a descrição `"My demonstration repository"` e uma tag com uma chave nomeada `Team` com o valor de `Saanvi`, use esse comando.

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

 Note

O campo de descrição exibe o Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o `GetRepository` ou `BatchGetRepositories` APIs e planeja exibir o

campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

3. Se houver êxito, o comando gerará como saída um objeto `repositoryMetadata` com as seguintes informações:
 - A descrição (`repositoryDescription`).
 - O ID exclusivo gerado pelo sistema (`repositoryId`).
 - O nome (`repositoryName`).
 - O ID da conta da Amazon Web Services associada ao CodeCommit repositório (`accountId`).

Veja a seguir um exemplo de saída com base no comando do exemplo anterior.

```
{
  "repositoryMetadata": {
    "repositoryName": "MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "lastModifiedDate": 1446071622.494,
    "repositoryDescription": "My demonstration repository",
    "cloneUrlHttp": "https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "defaultBranch": main,
    "kmsKeyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "creationDate": 1446071622.494,
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
    "accountId": "111111111111"
  }
}
```

Note

As tags que foram adicionadas quando o repositório foi criado não são retornadas na saída. Para visualizar uma lista de tags associadas a um repositório, execute o comando [list-tags-for-resource](#).

4. Anote o nome e o ID do CodeCommit repositório. Você precisa que eles monitorem e alterem as informações sobre o CodeCommit repositório, especialmente se você usar AWS CLI.

Se você esquecer o nome ou o ID, siga as instruções em [Exibir detalhes CodeCommit do repositório \(AWS CLI\)](#).

Depois de criar um repositório, é possível conectar-se a ele e começar a adicionar código. Para obter mais informações, consulte [Conexão a um repositório](#). Você também pode adicionar seu repositório a um pipeline de entrega contínua. Para obter mais informações, consulte [Passo-a-passo de pipeline simples](#).

Conecte-se a um AWS CodeCommit repositório

Quando você se conecta a um CodeCommit repositório pela primeira vez, normalmente clona seu conteúdo na sua máquina local. Você também pode [adicionar](#) e [editar arquivos](#) em um repositório diretamente do CodeCommit console. Ou se você já tem um repositório local, pode adicionar um repositório do CodeCommit como remoto. Este tópico fornece instruções para se conectar a um CodeCommit repositório. Se você quiser migrar um repositório existente para CodeCommit, consulte [Migrar para o CodeCommit](#).

Note

Dependendo do seu uso, você poderá ser cobrado pela criação ou acesso a um repositório. Para obter mais informações, consulte [Preços](#) na página de informações CodeCommit do produto.

Tópicos

- [Pré-requisitos para se conectar a um repositório CodeCommit](#)
- [Conecte-se ao CodeCommit repositório clonando o repositório](#)
- [Conectar um repositório local ao repositório CodeCommit](#)

Pré-requisitos para se conectar a um repositório CodeCommit

Antes de clonar um CodeCommit repositório ou conectar um repositório local a um repositório: CodeCommit

- Você deve ter configurado o seu computador local com o software e as configurações necessárias para se conectar ao CodeCommit. Isso inclui instalar e configurar o Git. Para obter mais informações, consulte [Configurando](#) e [Começando com o Git e AWS CodeCommit](#).
- Você deve ter a URL do clone do CodeCommit repositório ao qual deseja se conectar. Para obter mais informações, consulte [Visualizar detalhes de repositórios](#).

Se você ainda não criou um CodeCommit repositório, siga as instruções em [Criar um repositório](#), copie a URL do clone do CodeCommit repositório e retorne a esta página.

Se você tem um CodeCommit repositório, mas não sabe o nome dele, siga as instruções em [Visualizar detalhes de repositórios](#).

- Você deve ter um local em sua máquina local para armazenar uma cópia local do CodeCommit repositório ao qual você se conecta. (Essa cópia local do CodeCommit repositório é conhecida como repositório local.) Em seguida, alterne para os comandos do Git e execute-os a partir daquele local. Por exemplo, você pode usar /tmp (para Linux, macOS ou Unix) ou c:\temp (para Windows) se estiver fazendo um clone temporário para fins de teste. Este é o caminho de diretório usado nesses exemplos.

Note

Você pode usar o diretório que desejar. Se você estiver clonando um repositório para uso em longo período, considere criar o clone de um diretório em funcionamento e não de um usado para arquivos temporários. Se você estiver usando um outro diretório do /tmp ou c:\temp, certifique-se de substituí-lo pelo nosso diretório quando seguir essas instruções.

Conecte-se ao CodeCommit repositório clonando o repositório

Se você ainda não tiver um repositório local, siga as etapas deste procedimento para clonar o CodeCommit repositório em sua máquina local.

1. Conclua os pré-requisitos, incluindo [Configurando](#).

Important

Se você não tiver concluído a configuração, não poderá se conectar ou clonar o repositório.

2. No diretório /tmp ou no c:\temp, use o Git para executar o comando clone. Os exemplos a seguir mostram como clonar um repositório chamado *MyDemoRepo* na região Leste dos EUA (Ohio).

Para HTTPS usando [credenciais do Git](#) ou o assistente de credencial incluído na AWS CLI:

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

Para HTTPS usando [git-remote-codecommit](#), assumindo o perfil padrão e a Região da AWS configurados na AWS CLI:

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

Para o SSH:

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

Neste exemplo, `git-codecommit.us-east-2.amazonaws.com` é o ponto de conexão do Git para a região Leste dos EUA (Ohio) em que o repositório existe, `MyDemoRepo` representa o nome do seu CodeCommit repositório e `my-demo-repo` representa o nome do diretório que o Git cria no diretório ou no diretório. /tmp c:\temp Para obter mais informações sobre Regiões da AWS esse suporte CodeCommit e as conexões Git para eles Regiões da AWS, consulte.

[Regiões e endpoints de conexão do Git](#)

Note

Ao usar SSH em sistemas operacionais Windows para clonar um repositório, talvez você precise adicionar o ID da chave SSH à string de conexão da seguinte forma:

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

Para obter mais informações, consulte [Para conexões SSH no Windows](#) e [Solução de problemas](#).

Depois que o Git cria o diretório, ele puxa uma cópia do seu CodeCommit repositório para o diretório recém-criado.

Se o CodeCommit repositório for novo ou estiver vazio, você verá uma mensagem informando que está clonando um repositório vazio. Isso é esperado.

 Note

Se você receber um erro informando que o Git não consegue encontrar o CodeCommit repositório ou que você não tem permissão para se conectar ao CodeCommit repositório, certifique-se de preencher os [pré-requisitos](#), incluindo a atribuição de permissões ao usuário do IAM e a configuração de suas credenciais de usuário do IAM para o Git e na máquina local. Além disso, confirme se você especificou o nome correto do repositório.

Depois de conectar com sucesso seu repositório local ao seu CodeCommit repositório, agora você está pronto para começar a executar comandos Git a partir do repositório local para criar commits, branches e tags e enviar e extrair do repositório. CodeCommit

Conectar um repositório local ao repositório CodeCommit

Conclua as etapas a seguir se você já tiver um repositório local e quiser adicionar um repositório do CodeCommit como remoto. Se você já tem um repositório remoto e deseja enviar seus commits para CodeCommit e para esse outro repositório remoto, siga as etapas em. [Confirmações de push em dois repositórios](#)

1. Conclua os [pré-requisitos](#).
2. No prompt de comando ou terminal, alterne para o diretório do repositório local e execute o git remote add comando para adicionar o CodeCommit repositório como um repositório remoto para o repositório local.

Por exemplo, o comando a seguir adiciona o controle remoto apelidado de **origin** `https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo`:

Para HTTPS:

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

Para o SSH:

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

Esse comando não retorna nada.

3. Para verificar se você adicionou o CodeCommit repositório como remoto para seu repositório local, execute o `git remote -v` comando, que deve criar uma saída semelhante à seguinte:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para o 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)
```

Depois de conectar com sucesso seu repositório local ao seu CodeCommit repositório, você estará pronto para começar a executar comandos Git a partir do repositório local para criar commits, branches e tags, além de enviar e extrair do repositório. CodeCommit

Compartilhar um AWS CodeCommit repositório

Depois de criar um CodeCommit repositório, você pode compartilhá-lo com outros usuários. Primeiro, determine se você usará um acesso federado, credenciais temporárias ou um provedor de identidade da web, como o IAM Identity Center, ao acessar CodeCommit, ou se deseja usar credenciais do Git ou pares de chaves SSH com usuários do IAM. Se você estiver usando o

primeiro, precisará configurar usuários, acesso e permissões para seu provedor de identidade e, em seguida, fornecer instruções para seus usuários usarem o git-remote-codecommit. Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#) e [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#).

Você não pode usar credenciais do Git ou pares de chaves SSH com acesso federado ou provedores de identidade, mas muitos IDEs funcionam melhor com essas credenciais. Primeiro, decida qual protocolo (HTTPS ou SSH) recomendar aos usuários ao clonar e usar um cliente do Git ou um IDE para conectar ao seu repositório. Em seguida, envie o URL e as informações de conexão para os usuários com quem deseja compartilhar o repositório. Dependendo dos seus requisitos de segurança, compartilhar um repositório também pode exigir a criação de um grupo do IAM, a aplicação de políticas gerenciadas para esse grupo, a edição de políticas do IAM para refinar o acesso ou a criação e uso de perfis do IAM.

 Note

Depois de conceder aos usuários acesso ao console do repositório, eles poderão adicionar ou editar arquivos diretamente no console sem a necessidade de configurar um cliente Git ou outra conexão. Para obter mais informações, consulte [Criar ou adicionar um arquivo para um repositório do AWS CodeCommit](#) e [Editar o conteúdo de um arquivo em um AWS CodeCommit repositório](#).

Essas instruções foram escritas presumindo que você já tenha concluído as etapas em [Configurando](#) e [Criar um repositório](#).

 Note

Dependendo do uso, você poderá ser cobrado pela criação ou acesso a um repositório. Para obter mais informações, consulte [Preços](#) na página de informações CodeCommit do produto.

Tópicos

- [Escolha do protocolo de conexão para compartilhar com os usuários](#)
- [Criar políticas do IAM para o seu repositório](#)
- [Criar um grupo do IAM para usuários do repositório](#)
- [Compartilhar as informações de conexão com os usuários](#)

Escolha do protocolo de conexão para compartilhar com os usuários

Quando você cria um repositório no CodeCommit, dois endpoints são gerados: um para conexões HTTPS e outro para conexões SSH. Ambos fornecem conexões seguras em uma rede. Os usuários podem usar um dos dois protocolos. Ambos os endpoints se mantêm ativos, independentemente do protocolo que você recomenda aos usuários.

As conexões HTTPS exigem:

- As credenciais do Git, que os usuários do IAM podem gerar para si próprios no IAM. As credenciais do Git são o método mais fácil para os usuários do repositório configurarem e usarem.
- Uma chave de AWS acesso ou função a ser assumida, que os usuários do repositório devem configurar em seu perfil de credencial. É possível configurar `git-remote-codecommit` (recomendado) ou o assistente de credencial incluído na AWS CLI. Esses são os únicos métodos disponíveis para a conta raiz ou usuários federados.

Conexões SSH exigem que seus usuários:

- Gerem um par de chaves pública e privada.
- Armazenem a chave pública.
- Associar a chave pública ao usuário do IAM.
- Configurem seu arquivo de hosts conhecidos no computador local.
- Criem e mantenham um arquivo de configuração nos computadores locais.

Como esse é um processo de configuração mais complexo, recomendamos que você escolha as credenciais HTTPS e Git para conexões com CodeCommit.

Para obter mais informações sobre HTTPS, SSH, Git, `git-remote-codecommit` e repositórios remotos, consulte [Configurando](#), [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#) ou consulte a documentação do Git.

Note

Embora o Git seja compatível com uma variedade de protocolos de conexão, o CodeCommit não é compatível com protocolos inseguros, como o protocolo local ou HTTP genérico.

Criar políticas do IAM para o seu repositório

AWS fornece três políticas gerenciadas no IAM para CodeCommit. Essas políticas não podem ser editadas e aplicadas a todos os repositórios associados à sua conta da Amazon Web Services. No entanto, é possível usá-las como modelos para criar suas próprias políticas gerenciadas personalizadas que se aplicam somente ao repositório que deseja compartilhar. A política gerenciada pelo cliente pode ser aplicada especificamente ao repositório que você deseja compartilhar. Para obter mais informações, consulte [Políticas gerenciadas](#) e [Grupos e usuários do IAM](#).

Tip

Para um controle mais refinado do acesso ao repositório, é possível criar mais de uma política gerenciada pelo cliente e aplicá-las a diferentes grupos e usuários do IAM.

Para obter informações sobre como analisar o conteúdo de políticas gerenciadas e como usar políticas para criar e aplicar permissões, consulte [Autenticação e controle de acesso para AWS CodeCommit](#).

Criação de uma política gerenciada por cliente para seu repositório

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Na área de navegação do Painel, escolha Políticas e, em seguida, Create Policy.
3. Na página Criar políticas, escolha Importar política gerenciada.
4. Na página Importar políticas gerenciadas, em Filtrar políticas, insira o **AWSCodeCommitPowerUser**. Selecione o botão ao lado do nome da política e escolha Importar.
5. Na página Create policy (Criar política), escolha JSON. Substitua a parte “*” da Resource linha para CodeCommit ações pelo Amazon Resource Name (ARN) do CodeCommit repositório, conforme mostrado aqui:

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

 Tip

Para encontrar o ARN do CodeCommit repositório, acesse o CodeCommit console, escolha o nome do repositório na lista e escolha Configurações. Para obter mais informações, consulte [Visualizar detalhes de repositórios](#).

Se você quiser que essa política seja aplicada a mais de um repositório, adicione cada um como um recurso especificando o ARN dele. Inclua uma vírgula entre cada instrução de recurso, conforme mostrado aqui:

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

Ao concluir a edição, selecione Revisar política.

- Na página Revisar Política, em Nome, insira um novo nome para a política (por exemplo, *AWSCodeCommitPowerUser-MyDemoRepo*). Como opção, você pode fornecer uma descrição para essa política.
- Escolha Create Policy.

Criar um grupo do IAM para usuários do repositório

Para gerenciar o acesso ao repositório, crie um grupo do IAM para os usuários, adicione os usuários do IAM ao grupo e anexe a política gerenciada pelo cliente que você criou na etapa anterior. Como alternativa, você pode criar um perfil com uma política gerenciada pelo cliente anexada e fazer com que os usuários assumam esse perfil.

Se você usa SSH, deve anexar outra política gerenciada ao IAMUser SSHKeys grupo, a política gerenciada do IAM, que permite aos usuários fazer upload da chave pública SSH e associá-la ao usuário do IAM ao qual eles usam para se conectar. CodeCommit

- Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- Na área de navegação Painel, escolha Grupos e Criar novo grupo.

3. Na página Set Group Name (Definir nome de grupo), em Group Name (Nome do grupo), insira um nome para o grupo (por exemplo, *MyDemoRepoGroup*) e escolha Next Step (Próxima etapa). Considere incluir o nome do repositório como parte do nome do grupo.

 Note

Esse nome precisa ser exclusivo em toda a conta da Amazon Web Services.

4. Selecione a caixa ao lado da política gerenciada pelo cliente que você criou na seção anterior (por exemplo, *AWSCodeCommitPowerUser-MyDemoRepo*).
5. Na página Análise, escolha Criar grupo. O IAM criará esse grupo com as políticas especificadas já anexadas. O grupo aparecerá na lista de grupos associados à sua conta da Amazon Web Services.
6. Escolha seu grupo na lista.
7. Na página de resumo do grupo, escolha a guia Usuários e Adicionar usuários ao grupo. Na lista que mostra todos os usuários associados à sua conta da Amazon Web Services, selecione as caixas ao lado dos usuários aos quais você deseja permitir acesso ao CodeCommit repositório e escolha Adicionar usuários.

 Tip

Você pode usar a caixa Pesquisar para encontrar rapidamente usuários por nome.

8. Depois de adicionar os usuários, feche o console do IAM.

Compartilhar as informações de conexão com os usuários

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), escolha o repositório que deseja atualizar.
4. Em Clone URL (Clonar URL), escolha o protocolo que você deseja que os usuários usem. Isso copia a URL clonada para o protocolo de conexão.

5. Envie aos usuários o URL do clone junto com quaisquer outras instruções, como instalar o AWS CLI, configurar um perfil ou instalar o Git. Não se esqueça de incluir as informações de configuração para o protocolo de conexão (por exemplo, HTTPS).

O exemplo de e-mail a seguir fornece informações para usuários que se conectam ao MyDemoRepo repositório com o protocolo de conexão HTTPS e credenciais Git na região Leste dos EUA (Ohio) (us-east-2). Este e-mail foi escrito presumindo que o usuário já instalou o Git e está familiarizado com o seu uso.

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](#).

Você pode encontrar instruções completas de configuração em [Configurando](#).

Configurando notificações para eventos em um repositório AWS CodeCommit

É possível configurar regras de notificação para um repositório para que os usuários do repositório recebam e-mails sobre os tipos de evento de repositório especificados. As notificações são enviadas quando os eventos correspondem às configurações da regra de notificação. É possível criar um tópico do Amazon SNS para usar em notificações ou usar um tópico existente na sua conta da

Amazon Web Services. Você pode usar o CodeCommit console e o AWS CLI para configurar as regras de notificação.

[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 none](#) [Select 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

[X](#) [Remove target](#)

[Add target](#)

[Create SNS topic](#)

[Cancel](#) [Submit](#)

Tópicos

- [Usar regras de notificação do repositório](#)
- [Criar uma regra de notificação](#)
- [Alterar ou desabilitar notificações](#)
- [Excluir notificações](#)

Usar regras de notificação do repositório

Configurar regras de notificação ajuda os usuários do repositório ao enviar e-mails quando alguém realiza uma ação que afeta outro usuário. Por exemplo, é possível configurar uma regra de notificação para enviar notificações quando são feitos comentários em confirmações. Nessa configuração, quando um usuário do repositório comenta em uma linha de código em uma confirmação, outros usuários do repositório recebem um e-mail. Eles podem fazer login e visualizar o comentário. Respostas a comentários também geram e-mails, assim usuários do repositório se mantêm informados.

As regras de notificação são diferentes dos acionadores do repositório e também são diferentes das notificações que você poderia configurar no CodeCommit console antes de 5 de novembro de 2019.

- Embora você possa configurar um gatilho para usar o Amazon SNS a fim de enviar e-mails sobre alguns eventos de repositório, esses eventos estão limitados a eventos operacionais, como criação de ramificações e envio de código via push para uma ramificação. Os acionadores não usam regras de eventos para avaliar CloudWatch eventos do repositório. Eles são mais limitados em escopo. Para obter mais informações sobre o uso de triggers, consulte [Gerenciar triggers para um repositório](#).
- As notificações configuradas antes de 5 de novembro de 2019 tinham menos tipos de eventos disponíveis e não podiam ser configuradas para integração às salas de conversa do Amazon Chime ou canais do Slack. Você pode continuar a usar notificações configuradas antes de 5 de novembro de 2019, mas não é possível criar notificações desse tipo. Em vez disso, crie e use regras de notificação. Recomendamos usar regras de notificação e desativar ou excluir notificações criadas antes de 5 de novembro de 2019. Para obter mais informações, consulte [Criar uma regra de notificação](#) e [Excluir notificações](#).

Criar uma regra de notificação

É possível usar regras de notificação para notificar os usuários sobre alterações importantes, como quando uma solicitação pull é criada em um repositório. As regras de notificação especificam os

eventos e o tópico do Amazon SNS usados para enviar notificações. Para obter mais informações, consulte [O que são notificações?](#)

 Note

Esse atributo não está disponível na região Europa (Milão). Para saber como configurar notificações na experiência disponível nessa região, consulte [Configurar notificações do repositório](#).

Você pode usar o console ou o AWS CLI para criar regras de notificação para AWS CodeCommit.

Como criar uma regra de notificação (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeCommit console em <https://console.aws.amazon.com/codecommit/>.
2. Selecione Repositories (Repositórios) e escolha um repositório no qual você deseja adicionar regras de notificação.
3. Na página do repositório, selecione Notify (Notificar) e Create notification rule (Criar regra de notificação). Também é possível acessar a página Settings (Configurações) do repositório e selecionar Create notification rule (Criar regra de notificação).
4. Em Notification name (Nome da notificação), insira um nome para a regra.
5. Em Tipo de detalhe, escolha Básico se quiser que somente as informações fornecidas à Amazon sejam EventBridge incluídas na notificação. Escolha Completo se quiser incluir informações fornecidas à Amazon EventBridge e informações que possam ser fornecidas pelo gerenciador de notificações CodeCommit ou pelo gerenciador de notificações.

Para obter mais informações, consulte [Noções básicas sobre o conteúdo e a segurança de notificações](#).

6. Em Events that trigger notifications (Eventos que acionam notificações), selecione os eventos para os quais você deseja enviar notificações. Para obter mais informações, consulte [Eventos para regras de notificação em repositórios](#).
7. Em Targets (Destinos), siga um destes procedimentos:
 - Se você já tiver configurado um recurso a ser usado com notificações, em Escolher tipo de destino, escolha Amazon Q Developer em aplicações de chat (Slack) ou Tópico do SNS. Em

Escolher destino, escolha o nome do cliente (para um cliente do Slack configurado no Amazon Q Developer em aplicações de chat) ou o nome do recurso da Amazon (ARN) do tópico do Amazon SNS (para tópicos do Amazon SNS já configurados com a política necessária para notificações).

- Se você não configurou um recurso para usar com notificações, escolha Create target (Criar destino) e selecione SNS topic (Tópico do SNS). Forneça um nome para o tópico após codestar-notifications- e escolha Create (Criar).

Note

- Ao criar o tópico do Amazon SNS como parte da criação da regra de notificação, a política que permite ao recurso publicar eventos no tópico é aplicada para você. O uso de um tópico criado para regras de notificação ajuda a garantir que você inscreva somente os usuários para os quais deseja enviar notificações sobre esse recurso.
- Você não pode criar um Amazon Q Developer no cliente de aplicações de chat como parte da criação de uma regra de notificação. Se escolher o Amazon Q Developer em aplicações de chat (Slack), você verá um botão direcionando para configurar um cliente no Amazon Q Developer em aplicações de chat. A escolha dessa opção abre o Amazon Q Developer no console de aplicações de chat. Para obter mais informações, consulte [Configurar integrações entre notificações e o Amazon Q Developer em aplicações de chat](#).
- Se quiser usar um tópico do Amazon SNS existente como destino, você deverá adicionar a política necessária para o AWS CodeStar Notifications, além de quaisquer outras políticas que possam existir para esse tópico. Para obter mais informações, consulte [Configurar tópicos do Amazon SNS existentes para notificações](#) e [Noções básicas sobre conteúdos de notificações e segurança](#).

8. Para concluir a criação da regra, escolha Submit (Enviar).
9. Você precisa inscrever os usuários no tópico do Amazon SNS para a regra antes que eles possam receber notificações. Para obter mais informações, consulte [Inscrever usuários em tópicos do Amazon SNS que são destinos](#). Também é possível configurar a integração entre as notificações e o Amazon Q Developer em aplicações de chat para enviar notificações a salas de chat do Amazon Chime. Consulte mais informações em [Configurar integrações entre notificações e o Amazon Q Developer em aplicações de chat](#).

Criar uma regra de notificação (AWS CLI)

1. Em um terminal ou prompt de comando, execute o comando `create-notification-rule` para gerar o esqueleto JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

É possível nomear o arquivo como você quiser. Neste exemplo, o nome do arquivo é *rule.json*.

2. Abra o arquivo JSON em um editor de texto simples e edite-o para incluir o recurso, os tipos de evento e o destino que você deseja para a regra. O exemplo a seguir mostra uma regra de notificação com **MyNotificationRule** o nome de um repositório nomeado *MyDemoRepo* em uma AWS conta com o ID. *123456789012* As notificações com o tipo de detalhe completo são enviadas para um tópico do Amazon SNS chamado *MyNotificationTopic* quando ramificações e tags são criadas:

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeId": [ "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"  
}
```

Salve o arquivo.

3. Usando o arquivo que você acabou de editar, no terminal ou na linha de comando, execute o comando `create-notification-rule` novamente para criar a regra de notificação:

```
aws codestar-notifications create-notification-rule --cli-input-json
file://rule.json
```

4. Se for bem-sucedido, o comando retornará o ARN da regra de notificação, semelhante ao seguinte:

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

Alterar ou desabilitar notificações

Você pode usar o AWS CodeCommit console para alterar a forma como as notificações criadas antes de 5 de novembro de 2019 são configuradas, incluindo os tipos de eventos que enviam e-mails aos usuários e o tópico do Amazon SNS usado para enviar e-mails sobre o repositório. Também é possível usar o console do CodeCommit para gerenciar a lista de endereços de e-mail e endpoints inscritos no tópico ou para desativar notificações.

Para alterar configurações de notificação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja configurar as notificações criadas antes de 5 de novembro de 2019.
3. No painel de navegação, escolha Settings e Notifications. Se você vir um banner informando que tem notificações em vez de regras de notificação, selecione Manage existing notifications (Gerenciar notificações existentes).
4. Escolha Editar.
5. Faça suas alterações e escolha Salvar.

Desativar notificações é uma maneira fácil de impedir temporariamente que os usuários recebam e-mails sobre eventos de repositório.

Para excluir permanentemente uma notificação criada antes de 5 de novembro de 2019, siga as etapas em [Excluir notificações](#).

Como desativar notificações

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja desativar as notificações.
3. No painel de navegação, escolha Settings e Notifications. Selecione Manage existing notifications (Gerenciar notificações existentes).
4. Selecione Edit (Editar) e, em Event status (Status do evento), use o controle deslizante para desativar Enable notifications (Ativar notificações). Escolha Salvar.
5. O status do evento muda para Disabled (Desativado). Nenhum e-mail sobre eventos é enviado. Quando você desativa as notificações, a regra de CloudWatch Eventos do repositório é desativada automaticamente. Não altere manualmente seu status no console de CloudWatch Eventos.

Excluir notificações

Se você não quiser mais usar as notificações criadas para um repositório antes de 5 de novembro de 2019, você pode excluir a regra Amazon CloudWatch Events associada à notificação. Isso excluirá automaticamente a notificação. Ela não exclui as assinaturas ou o tópico do SNS usado para notificações.

Note

Se você alterar o nome de um repositório no console, as notificações criadas antes de 5 de novembro de 2019 continuarão a funcionar sem modificação. No entanto, se você alterar o nome do seu repositório a partir da linha de comando ou usando a API, as notificações deixarão de funcionar. A maneira mais fácil de restaurar notificações é excluir as configurações de notificação e depois voltar a defini-las.

Para excluir configurações de notificação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório do qual deseja excluir notificações criadas antes de 5 de novembro de 2019.

3. No painel de navegação, escolha Settings e Notifications. Se você vir um banner informando que tem notificações em vez de regras de notificação, selecione Manage existing notifications (Gerenciar notificações existentes).
4. Na regra do CloudWatch evento, copie o nome da regra que foi criada para a notificação.
5. Faça login no Console de gerenciamento da AWS e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
6. Em Events (Eventos), selecione Rules (Regras). Em Name (Nome), cole o nome da regra criada para a notificação. Escolha a regra e, em Actions (Ações), selecione Delete (Excluir).
7. (Opcional) Para alterar ou excluir o tópico do Amazon SNS usado para notificações depois de excluir as configurações de notificação, acesse o console do Amazon SNS em v3/home. <https://console.aws.amazon.com/sns/> Para obter mais informações, consulte [Limpeza](#) no [Guia do desenvolvedor do Amazon Simple Notification Service](#).

Marcando repositórios em AWS CodeCommit

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. AWS as tags são diferentes das tags do Git, que podem ser aplicadas aos commits. Cada AWS tag tem duas partes:

- Uma chave de tag (por exemplo CostCenter, Environment, Project ou Secret). As chaves de tag diferenciam maiúsculas de minúsculas
- Um campo opcional conhecido como um valor de tag (por exemplo, 111122223333, Production ou um nome de equipe). Omitir o valor da tag é o mesmo que usar uma string vazia. Assim como as chaves de tag, os valores de tag diferenciam maiúsculas de minúsculas.

Juntos, esses são conhecidos como pares de chave-valor. Para limites do número de tags que é possível ter em um repositório e restrições sobre valores e chaves de tags, consulte [Limites](#).

As tags ajudam você a identificar e organizar seus AWS recursos. Muitos AWS serviços oferecem suporte à marcação, então você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados. Por exemplo, você pode atribuir a mesma tag a um CodeCommit repositório que você atribui a um bucket do Amazon S3. Para obter mais informações sobre estratégias de marcação, consulte Recursos de [marcação AWS](#).

Em CodeCommit, o recurso principal é um repositório. Você pode usar o CodeCommit console, o AWS CLI CodeCommit APIs, ou AWS SDKs para adicionar, gerenciar e remover tags de um

repositório. Além de identificar, organizar e acompanhar seu repositório com tags, você pode usar tags em políticas do IAM para ajudar a controlar quem pode visualizar e interagir com seu repositório. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Tópicos

- [Adicionar uma tag a um repositório](#)
- [Visualizar as tags de um repositório](#)
- [Editar tags de um repositório](#)
- [Remover uma tag de um repositório](#)

Adicionar uma tag a um repositório

Adicionar tags a um repositório pode ajudar você a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Primeiro adicione uma ou mais tags (pares chave-valor) a um repositório. Lembre-se de que há limites para o número de tags que é possível ter em um repositório. Há restrições com relação aos caracteres que é possível usar nos campos chave e valor. Para saber mais, consulte [Limites](#). Depois que você tiver tags, poderá criar políticas do IAM para gerenciar o acesso a repositórios com base nessas tags. Você pode usar o CodeCommit console ou o AWS CLI para adicionar tags a um repositório.

Important

Adicionar tags a um repositório pode afetar o acesso ele. Para adicionar uma tag a um repositório, revise as políticas do IAM que podem usar tags para controlar o acesso a recursos, como repositórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Para obter mais informações sobre como adicionar tags a um repositório ao criá-lo, consulte [Criar um repositório \(console\)](#).

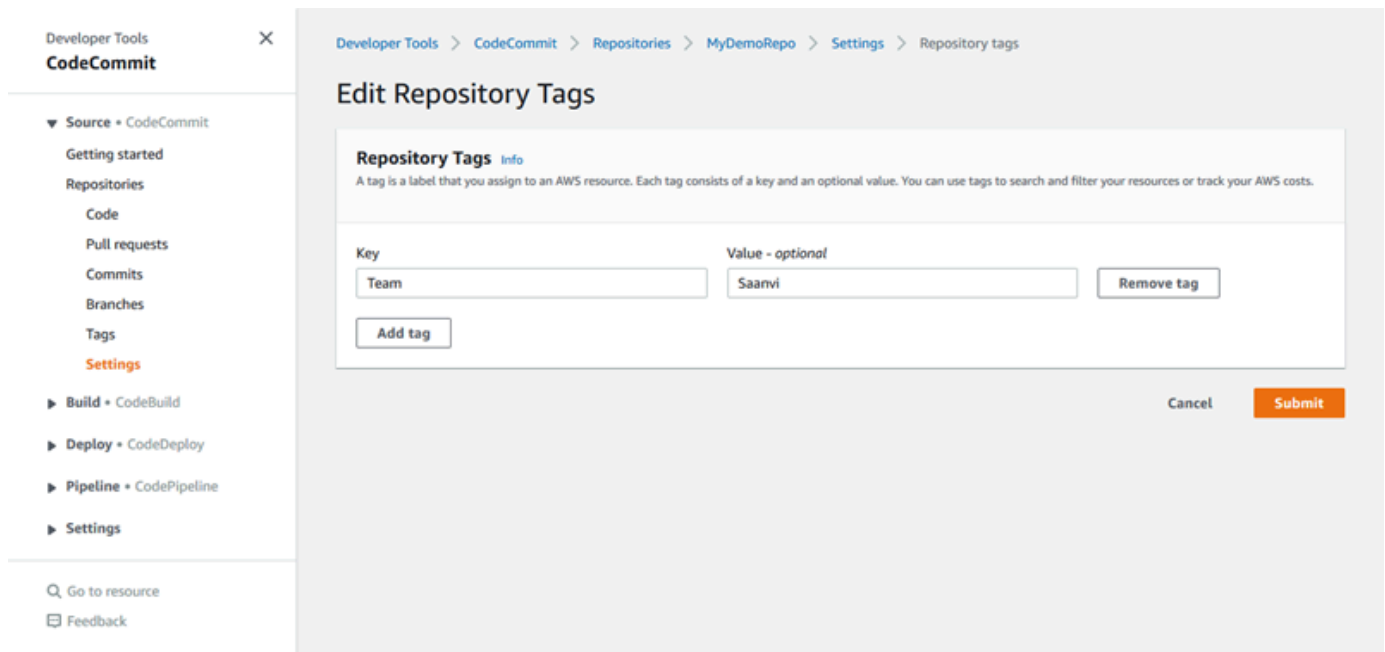
Tópicos

- [Adicionar uma tag a um repositório \(console\)](#)
- [Adicionar uma tag a um repositório \(AWS CLI\)](#)

Adicionar uma tag a um repositório (console)

Você pode usar o CodeCommit console para adicionar uma ou mais tags a um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório no qual deseja adicionar tags.
3. No painel de navegação, selecione Configurações. Selecione Repository tags (Tags de repositório).
4. Se nenhuma tag tiver sido adicionada ao repositório, selecione Add tag (Adicionar tag). Caso contrário, selecione Edit (Editar) e Add tag (Adicionar tag).
5. Em Chave, insira um nome para a tag. É possível adicionar um valor opcional para a tag em Valor.



6. (Opcional) Para adicionar outra tag, selecione Adicionar tag novamente.
7. Quando terminar de adicionar tags, selecione Enviar.

Adicionar uma tag a um repositório (AWS CLI)

Siga estas etapas para usar o AWS CLI para adicionar uma tag a um CodeCommit repositório. Para adicionar uma tag a um repositório ao criá-lo, consulte [Criar um repositório \(AWS CLI\)](#).

Nestas etapas, partimos do princípio de que você já instalou uma versão recente da AWS CLI ou atualizou para a versão atual. Para obter mais informações, consulte [Instalar a AWS Command Line Interface](#).

No terminal ou na linha de comando, execute o comando `tag-resource`, especificando o nome de recurso da Amazon (ARN) do repositório no qual você deseja adicionar tags e a chave e o valor da tag que você deseja adicionar. Você pode adicionar mais de uma tag a um repositório. Por exemplo, para marcar um repositório nomeado *MyDemoRepo* com duas tags, uma chave de tag *Status* com o valor de tag de *Secret* e uma chave de tag *Team* com o valor de tag de *Saanvi*:

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Status=Secret,Team=Saanvi
```

Se houver êxito, o comando não retornará nada.

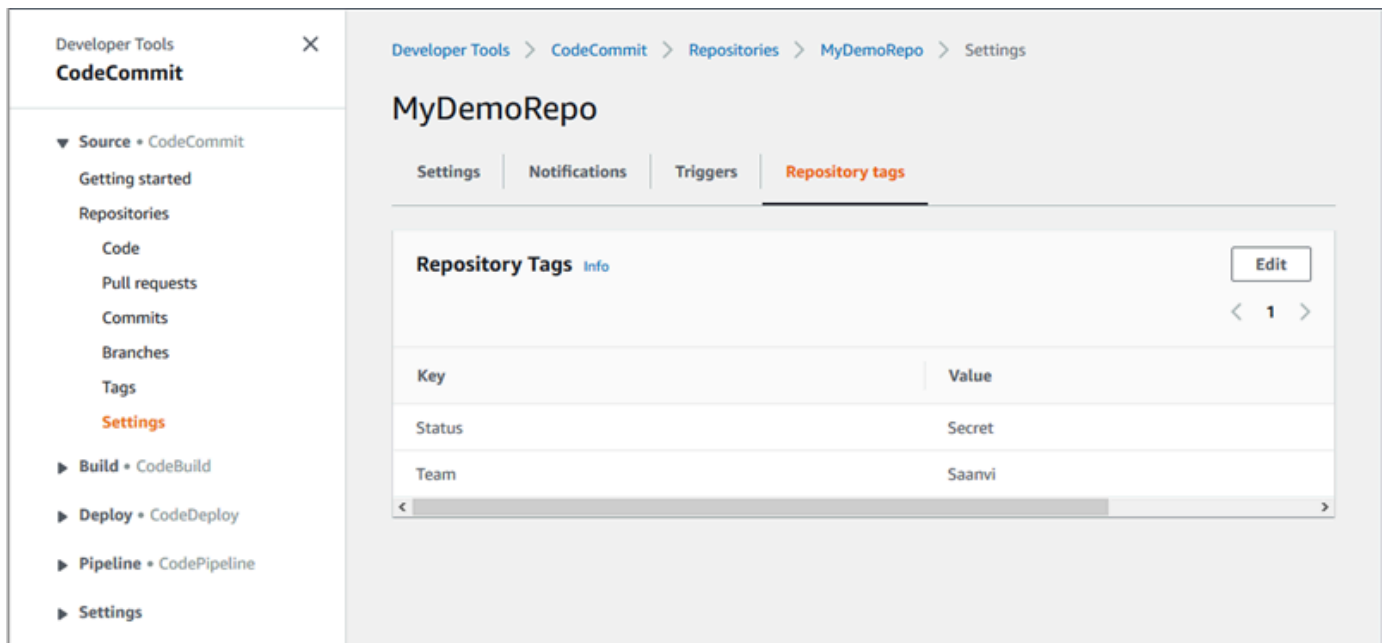
Visualizar as tags de um repositório

As tags podem ajudar você a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Para obter mais informações sobre estratégias de marcação, consulte Recursos de [marcação AWS](#). Para obter exemplos de políticas de acesso baseadas em tags, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Visualizar as tags de um repositório (console)

Você pode usar o CodeCommit console para visualizar as tags associadas a um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja visualizar as tags.
3. No painel de navegação, selecione Configurações. Selecione Repository tags (Tags de repositório).



Visualizar as tags de um repositório (AWS CLI)

Siga estas etapas para usar o AWS CLI para visualizar as AWS tags de um CodeCommit repositório. Se não foram adicionadas tags, a lista retornará vazia.

No terminal ou na linha de comando, execute o comando `list-tags-for-resource`. Por exemplo, para visualizar uma lista de chaves de tag e valores de tag para um repositório chamado *MyDemoRepo* com o 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 houver êxito, o comando retornará informações semelhantes às seguintes:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "Saanvi"
  }
}
```


Editar tags de um repositório

É possível alterar o valor de uma tag associada a um repositório. Também é possível alterar o nome da chave, o que é equivalente a remover a tag atual e adicionar outra com o novo nome e o mesmo valor da outra chave. Lembre-se de que há limites com relação aos caracteres que podem ser usados nos campos chave e valor. Para saber mais, consulte [Limites](#).

Important

Editar as tags de um repositório pode afetar o acesso a ele. Antes de editar o nome (chave) ou o valor de uma tag de um repositório, revise as políticas do IAM que podem usar a chave ou o valor de uma tag para controlar o acesso a recursos, como repositórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Editar uma tag de um repositório (console)

Você pode usar o CodeCommit console para editar as tags associadas a um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório no qual deseja editar tags.
3. No painel de navegação, selecione Configurações. Selecione Repository tags (Tags de repositório).
4. Escolha Editar.

5.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Settings > Repository tags

Edit Repository Tags

Repository Tags [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Each resource can have up to 50 tags. Keys cannot begin with "AWS:".

Key	Value - optional	
Status	Secret	Remove tag
Team	Saanvi	Remove tag

Add tag

Cancel Submit

Execute um destes procedimentos:

- Para alterar a tag, insira um novo nome em Chave. Alterar o nome da tag é equivalente a excluir uma tag e adicionar outra tag com o novo nome de chave.
- Para alterar o valor de uma tag, insira um novo valor. Se você deseja alterar o valor para nulo, exclua o valor atual e deixe o campo em branco.

6. Quando terminar de editar as tags, selecione Enviar.

Editar tags de um repositório (AWS CLI)

Siga estas etapas para usar o AWS CLI para atualizar uma tag para um CodeCommit repositório. Você pode alterar o valor para uma chave existente ou adicionar outra chave.

No terminal ou na linha de comando, execute o comando `tag-resource`, especificando o nome de recurso da Amazon (ARN) do repositório em que você deseja atualizar uma tag e especifique a chave e o valor da tag:

```
aws codecommit tag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tags Team=Li
```

Remover uma tag de um repositório

É possível excluir uma ou mais tags associadas a um repositório. A remoção de uma tag não exclui a tag de outros AWS recursos associados a essa tag.

Important

A exclusão de tags de um repositório pode afetar o acesso a ele. Antes de remover uma tag de um repositório, revise as políticas do IAM que podem usar a chave ou o valor de uma tag para controlar o acesso a recursos, como repositórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Remover uma tag de um repositório (console)

Você pode usar o CodeCommit console para remover a associação entre uma tag e um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório de onde deseja excluir tags.
3. No painel de navegação, selecione Configurações. Selecione Repository tags (Tags de repositório).
4. Escolha Editar.
5. Encontre a tag que deseja excluir e selecione Remove tag.
6. Quando terminar de excluir tags, selecione Enviar.

Remover uma tag de um repositório (AWS CLI)

Siga estas etapas para usar o AWS CLI para remover uma tag de um CodeCommit repositório. Remover uma tag não a exclui, mas simplesmente remove a associação entre a tag e o repositório.

Note

Se você excluir um CodeCommit repositório, todas as associações de tags serão removidas do repositório excluído. Não é necessário remover tags antes de excluir um repositório.

No terminal ou na linha de comando, execute o comando `untag-resource`, especificando o ARN do repositório no qual você deseja remover tags e a chave da tag que você deseja remover. Por exemplo, para remover uma tag em um repositório chamado *MyDemoRepo* com a chave *Status* da tag:

```
aws codecommit untag-resource --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo --tag-keys Status
```

Se houver êxito, o comando não retornará nada. Para verificar as tags associadas ao repositório, execute o comando `list-tags-for-resource`.

Gerenciar gatilhos para um repositório AWS CodeCommit

Você pode configurar um CodeCommit repositório para que envios de código ou outros eventos acionem ações, como enviar uma notificação do Amazon Simple Notification Service (Amazon SNS) ou invocar uma função no AWS Lambda. Você pode criar até 10 acionadores para cada CodeCommit repositório.

Os gatilhos costumam ser configurados para:

- Enviar e-mails para usuários inscritos sempre que alguém efetuar um push para o repositório.
- Notificar um sistema de desenvolvimento externo para que comece uma compilação após alguém efetuar um push para a ramificação principal do repositório.

Cenários como a notificação de um sistema de compilação externo exigem a gravação de uma função do Lambda para interagir com outras aplicações. O cenário de e-mail exige simplesmente a criação de um tópico do Amazon SNS.

Este tópico mostra como definir permissões que permitem CodeCommit acionar ações no Amazon SNS e no Lambda. Ele também inclui links para exemplos de criação, edição, teste e exclusão de gatilhos.

Tópicos

- [Crie o recurso e adicione permissões para CodeCommit](#)
- [Exemplo: criar um AWS CodeCommit gatilho para um tópico do Amazon SNS](#)
- [Exemplo: criar um AWS CodeCommit gatilho para uma AWS Lambda função](#)
- [Exemplo: criar um gatilho AWS CodeCommit para uma AWS Lambda função existente](#)

- [Editar gatilhos para um repositório AWS CodeCommit](#)
- [Acionadores de teste para um repositório AWS CodeCommit](#)
- [Excluir gatilhos de um repositório AWS CodeCommit](#)

Crie o recurso e adicione permissões para CodeCommit

Você pode integrar tópicos do Amazon SNS e funções do Lambda com gatilhos CodeCommit, mas você deve primeiro criar e depois configurar recursos com uma política que conceda CodeCommit as permissões para interagir com esses recursos. Você deve criar o recurso da Região da AWS mesma forma que o CodeCommit repositório. Por exemplo, se o repositório estiver no Leste dos EUA (Ohio) (us-east-2), o tópico do Amazon SNS ou a função do Lambda deverá estar no Leste dos EUA (Ohio).

- Para tópicos do Amazon SNS, você não precisa configurar políticas ou permissões adicionais do IAM se o tópico do Amazon SNS for criado usando a mesma conta do repositório. CodeCommit Você pode criar o CodeCommit gatilho assim que tiver criado e se inscrito no tópico do Amazon SNS.
 - Para obter mais informações sobre como criar um tópico no Amazon SNS, consulte [Documentação do Amazon SNS](#).
 - Para obter informações sobre como usar o Amazon SNS para enviar mensagens para filas do Amazon SQS, consulte [Enviar mensagens para filas do Amazon SQS](#) no Guia do desenvolvedor do Amazon SNS.
 - Para obter informações sobre como usar o Amazon SNS para invocar uma função do Lambda, consulte [Invocar funções do Lambda](#) no Guia do desenvolvedor do Amazon SNS.
- Você pode configurar funções do Lambda criando o gatilho no console do Lambda como parte da função. Esse é o método mais simples, pois os acionadores criados no console do Lambda incluem automaticamente as permissões necessárias para CodeCommit invocar a função Lambda. Se você criar o gatilho em CodeCommit, deverá incluir uma política para permitir CodeCommit a invocação da função. Para obter mais informações, consulte [Criar um gatilho para uma função do Lambda existente](#).

Exemplo: criar um AWS CodeCommit gatilho para um tópico do Amazon SNS

Você pode criar um gatilho para um CodeCommit repositório para que os eventos desse repositório acionem notificações de um tópico do Amazon Simple Notification Service (Amazon SNS). Você

pode criar um gatilho para um tópico do Amazon SNS para permitir que usuários se inscrevam em notificações sobre eventos de repositório, como a exclusão de ramificações. Você também pode aproveitar a integração dos tópicos do Amazon SNS com outros serviços, como o Amazon Simple Queue Service (Amazon SQS) e AWS Lambda

Note

- É necessário apontar o gatilho para um tópico do Amazon SNS, que é a ação realizada em resposta a eventos de repositório. Para obter mais informações sobre como criar e assinar tópicos do Amazon SNS, consulte [Conceitos básicos do Amazon Simple Notification Service](#).
- Os tópicos FIFO (primeiro a entrar, primeiro a sair) do Amazon SNS não são compatíveis com gatilhos. CodeCommit

Tópicos

- [Crie um gatilho para um tópico do Amazon SNS para um CodeCommit repositório \(console\)](#)
- [Crie um gatilho para um tópico do Amazon SNS para um CodeCommit repositório \(AWS CLI\)](#)

Crie um gatilho para um tópico do Amazon SNS para um CodeCommit repositório (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja criar gatilhos de eventos de repositório.
3. No painel de navegação do repositórios, selecione Settings (Configurações) e gatilhos (Gatilhos).
4. Selecione Create gatilho (Criar gatilho) e proceda da seguinte maneira:
 - Em Nome do acionador, insira um nome para o gatilho (por exemplo, *MyFirstTrigger*).
 - Em Eventos, selecione os eventos de repositório que acionarão o tópico do Amazon SNS para enviar notificações.

Se você selecionar All repository events, não poderá escolher nenhum outro evento. Para selecionar um subconjunto de eventos, remova All repository events e selecione um ou mais eventos na lista. Por exemplo, se você desejar que o gatilho seja executado somente quando

um usuário criar uma ramificação ou uma tag no repositório do CodeCommit remova All repository events (Todos os eventos de repositório) e selecione Create branch or tag (Criar ramificação ou tag).

- Se você deseja que o gatilho seja aplicado a todas as ramificações do repositório, em Branches (Ramificações), deixe a seleção em branco, pois essa opção padrão aplica o gatilho a todas as ramificações automaticamente. Se você quiser que esse gatilho seja aplicado somente a ramificações específicas, escolha até 10 nomes de ramificação na lista de ramificações do repositório.
- Em Escolher o serviço para usar, selecione Amazon SNS.
- Em Amazon SNS, escolha o nome de um tópico na lista ou insira o ARN do tópico.

 Note

Os tópicos FIFO (primeiro a entrar, primeiro a sair) do Amazon SNS não são compatíveis com gatilhos. CodeCommit Você deve escolher um tópico FIFO do Amazon SNS que tenha seu tipo definido como Padrão.

- Em Dados personalizados, forneça qualquer informação adicional que deseja incluir na notificação enviada pelo tópico do Amazon SNS (por exemplo, o nome de um canal IRC que os desenvolvedores usam quando debatem o desenvolvimento neste repositório). Este campo é uma string. Ele não pode ser usado para passar nenhum parâmetro dinâmico.
5. (Opcional) Selecione Test gatilho (Testar gatilho). Essa etapa ajuda você a confirmar se configurou corretamente o acesso entre CodeCommit e o tópico do Amazon SNS. Ele usa o tópico do Amazon SNS para enviar uma notificação de teste usando dados do repositório, se disponíveis. Se nenhum dado real estiver disponível, a notificação de teste conterá dados de exemplo.
 6. Selecione Create gatilho (Criar gatilho) para concluir a criação do gatilho.

Crie um gatilho para um tópico do Amazon SNS para um CodeCommit repositório ()AWS CLI

Você também pode usar a linha de comando para criar um gatilho para um tópico do Amazon SNS em resposta a eventos do CodeCommit repositório, como quando alguém envia um commit para o seu repositório.

Criar um gatilho para um tópico do Amazon SNS

1. Abra um editor de texto simples e crie um arquivo JSON que especifique:

- O nome do tópico do Amazon SNS.

Note

Os tópicos FIFO (primeiro a entrar, primeiro a sair) do Amazon SNS não são compatíveis com gatilhos. Você deve escolher um tópico FIFO do Amazon SNS que tenha seu tipo definido como Padrão.

- O repositório e as ramificações que deseja monitorar com o gatilho. (Se você não especificar nenhuma ramificação, o gatilho se aplicará a todas as ramificações no repositório.)
- Os eventos que ativam o gatilho.

Salve o arquivo.

Por exemplo, para criar um gatilho para um repositório chamado *MyDemoRepo* que publica todos os eventos do repositório em um tópico do Amazon SNS nomeado *MySNSTopic* para duas filiais e: *main preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyFirstTrigger",
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
      "customData": "",
      "branches": [
        "main", "preprod"
      ],
      "events": [
        "all"
      ]
    }
  ]
}
```


Deve haver um bloco do gatilho no JSON para cada gatilho de um repositório. Para criar mais de um gatilho para o repositório, inclua mais de um bloco de gatilho no JSON. Lembre-se de que todos os gatilhos criados no arquivo são para o repositório especificado. Você não pode criar gatilhos para vários repositórios em um único arquivo JSON. Por exemplo, se você desejasse criar dois gatilhos para um repositório, poderia criar um arquivo JSON com dois blocos de gatilhos. No exemplo a seguir, não há ramificação especificada para o segundo gatilho, então ele se aplica a todas as ramificações:

```
{
  "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"
      ]
    }
  ]
}
```

É possível criar gatilhos para eventos que você especifica, como quando uma confirmação é enviada por push para um repositório. Os tipos de eventos incluem:

- `all` para todos os eventos no repositório e nas ramificações especificados.

- `updateReference` para quando confirmações são enviadas por push para o repositório e a ramificações especificados.
- `createReference` para quando uma nova ramificação ou tag é criada no repositório especificado.
- `deleteReference` para quando uma nova ramificação ou tag é excluída no repositório especificado.

 Note

É possível usar mais de um tipo de evento em um gatilho. Contudo, se você especificar `all`, não poderá especificar outros eventos.

Para ver a lista completa dos tipos de eventos válidos em um terminal ou prompt de comando, insira `aws codecommit put-repository-triggers help`.

Além disso, é possível incluir uma string em `customData` (por exemplo, o nome de um canal IRC que os desenvolvedores usam quando debatem o desenvolvimento neste repositório). Este campo é uma string. Ele não pode ser usado para passar nenhum parâmetro dinâmico. Essa string é anexada como um atributo ao CodeCommit JSON retornado em resposta ao acionador.

2. (Opcional) No terminal ou no aviso de comando, execute o comando `test-repository-triggers`. Este teste usa dados de amostra do repositório (ou gera dados de amostra se não houver dados disponíveis) para enviar uma notificação para os assinantes do tópico do Amazon SNS. Por exemplo, o seguinte é usado para testar se o JSON no arquivo acionador chamado *trigger.json* é válido e CodeCommit pode ser publicado no tópico do Amazon SNS:

```
aws codecommit test-repository-triggers --cli-input-json file:///trigger.json
```

Se houver êxito, o comando retornará informações semelhantes às seguintes:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

3. Em um terminal ou prompt de comando, execute o comando `put-repository-triggers` para criar o gatilho no CodeCommit. Por exemplo, para usar um arquivo JSON chamado *trigger.json* para criar o acionador:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Esse comando retorna um [ID de configuração](#) semelhante ao seguinte:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

4. Para visualizar a configuração do gatilho, execute o comando `get-repository-triggers`, especificando o nome do repositório:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Esse comando retorna a estrutura de todos os gatilhos configurados para o repositório semelhante à seguinte::

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE",
  "triggers": [
    {
      "events": [
        "all"
      ],
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MySNSTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "Project ID 12345"
    }
  ]
}
```

5. Para testar a funcionalidade do próprio gatilho, crie e envie uma confirmação por push para o repositório onde você o configurou. Você deverá visualizar uma resposta do tópico do Amazon

SNS. Por exemplo, se você configurou o tópico do Amazon SNS para enviar um e-mail, você verá um e-mail do Amazon SNS na conta de e-mail inscrita no tópico.

Veja a seguir um exemplo de saída de um e-mail enviado pelo Amazon SNS em resposta a um envio para um CodeCommit repositório:

```
{
  "Records": [
    {
      "awsRegion": "us-east-2",
      "codecommit": {
        "references": [
          {
            "commit": "317f8570EXAMPLE",
            "created": true,
            "ref": "refs/heads/NewBranch"
          },
          {
            "commit": "4c925148EXAMPLE",
            "ref": "refs/heads/preprod",
          }
        ]
      },
      "eventId": "11111-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",
    }
  ]
}
```

Exemplo: criar um AWS CodeCommit gatilho para uma AWS Lambda função

Você pode criar um gatilho para um CodeCommit repositório para que os eventos no repositório invoquem uma função Lambda. Neste exemplo, você cria uma função Lambda que retorna a URL usada para clonar o repositório em um log da Amazon. CloudWatch

Tópicos

- [Criar a função do Lambda](#)
- [Veja o acionador da função Lambda no repositório AWS CodeCommit](#)

Criar a função do Lambda

Ao usar o console Lambda para criar a função, você também pode criar um CodeCommit gatilho para a função Lambda. As seguintes etapas incluem uma amostra de função do lambda. A amostra está disponível em duas linguagens: JavaScript e Python. A função retorna o URLs usado para clonar um repositório em um CloudWatch log.

Para criar uma função do lambda usando um esquema do Lambda

1. Faça login no Console de gerenciamento da AWS e abra o AWS Lambda console em <https://console.aws.amazon.com/lambda/>.
2. Na página Funções do Lambda, escolha Criar uma função. (Se você ainda não usou o Lambda, selecione Comece agora.)
3. Na página Create function, selecione Author from scratch. Em Nome da função, forneça um nome para a função, por exemplo *MyLambdaFunctionforCodeCommit*. Em Runtime (Tempo de execução), escolha o idioma que deseja usar para escrever a função e selecione Create function (Criar função).
4. Na guia Configuration (Configuração), selecione Add trigger (Adicionar trigger).
5. Na configuração do Trigger, escolha na CodeCommit lista suspensa de serviços.

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

- Em Nome do repositório, escolha o nome do repositório onde você deseja configurar um gatilho que usará a função do Lambda em resposta a eventos de repositório.

- Em Nome do acionador, insira um nome para o gatilho (por exemplo, *MyLambdaFunctionTrigger*).
- Em Eventos, escolha os eventos de repositório que acionarão a função do Lambda. Se você selecionar All repository events, não poderá escolher nenhum outro evento. Se desejar escolher um subconjunto de eventos, limpe All repository events e, em seguida, escolha os eventos que deseja na lista. Por exemplo, se você quiser que o gatilho seja executado somente quando um usuário criar uma tag ou uma ramificação no AWS CodeCommit repositório, remova Todos os eventos do repositório e escolha Criar ramificação ou tag.
- Se desejar que o gatilho se aplique a todas as ramificações do repositório, em Branches, selecione All branches. Caso contrário, escolha Specific branches. A ramificação padrão do repositório é adicionada por padrão. Você pode manter ou excluir essa ramificação da lista. Escolha até 10 nomes de ramificação na lista de ramificações de repositório.
- (Opcional) Em Dados personalizados, insira as informações que deseja incluir na função do Lambda (por exemplo, o nome de um canal IRC que os desenvolvedores usam quando debatem o desenvolvimento no repositório). Este campo é uma string. Ele não pode ser usado para passar nenhum parâmetro dinâmico.

Escolha Adicionar.

6. Na página Configuration (Configuração), em Function Code (Código de função), em Code entry type (Tipo de entrada de código), selecione Edit code inline (Editar código em linha). Em Runtime (Tempo de execução), selecione Node.js. Se você quiser criar uma função Python de exemplo, selecione Python.
7. Em Code entry type, selecione Edit code inline e substitua o código hello world por um dos seguintes exemplos.

Para Node.js:

```
import {
    CodeCommitClient,
    GetRepositoryCommand,
} from "@aws-sdk/client-codecommit";

const codecommit = new CodeCommitClient({ region: "your-region" });

/**
 * @param {{ Records: { codecommit: { references: { ref: string }[] },
    eventSourceARN: string }[] } event
```

```
*/
export const handler = async (event) => {
  // Log the updated references from the event
  const references = event.Records[0].codecommit.references.map(
    (reference) => reference.ref,
  );
  console.log("References:", references);

  // Get the repository from the event and show its git clone URL
  const repository = event.Records[0].eventSourceARN.split(":")[5];
  const params = {
    repositoryName: repository,
  };

  try {
    const data = await codecommit.send(new GetRepositoryCommand(params));
    console.log("Clone URL:", data.repositoryMetadata.cloneUrlHttp);
    return data.repositoryMetadata.cloneUrlHttp;
  } catch (error) {
    console.error("Error:", error);
    throw new Error(
      `Error getting repository metadata for repository ${repository}`,
    );
  }
};
```

Para Python:

```
import json
import boto3

codecommit = boto3.client("codecommit")

def lambda_handler(event, context):
    # Log the updated references from the event
    references = {
        reference["ref"]
        for reference in event["Records"][0]["codecommit"]["references"]
    }
    print("References: " + str(references))
```



```
# Get the repository from the event and show its git clone URL
repository = event["Records"][0]["eventSourceARN"].split(":")[5]
try:
    response = codecommit.get_repository(repositoryName=repository)
    print("Clone URL: " + response["repositoryMetadata"]["cloneUrlHttp"])
    return response["repositoryMetadata"]["cloneUrlHttp"]
except Exception as e:
    print(e)
    print(
        "Error getting repository {}. Make sure it exists and that your
repository is in the same region as this function.".format(
            repository
        )
    )
    raise e
```

8. Na guia Permissões, em Função de execução, selecione a função para abri-la no console do IAM. Edite a política anexada para adicionar a permissão `GetRepository` ao repositório que deve usar o trigger.

Veja o acionador da função Lambda no repositório AWS CodeCommit

Depois de criar a função do Lambda, será possível visualizar e testar o gatilho no AWS CodeCommit. O teste do gatilho executa a função em resposta aos eventos de repositório que você especificar.

Para visualizar e testar o gatilho da função do lambda

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório que onde deseja exibir gatilhos.
3. No painel de navegação do repositórios, selecione Settings (Configurações) e gatilhos (Gatilhos).
4. Analise a lista de gatilhos do repositório. Você deverá visualizar o gatilho que criou no console do Lambda. Selecione-o na lista e selecione Test trigger (Testar gatilho). Essa opção tenta chamar a função com os dados de exemplo sobre o seu repositório, incluindo o ID de confirmação mais recente dele. (Se não houver histórico de confirmações, serão gerados valores de exemplo que consistem em zeros.) Isso ajuda você a confirmar que você configurou corretamente o acesso entre AWS CodeCommit e a função Lambda.

5. Para verificar mais a funcionalidade do próprio gatilho, crie e envie uma confirmação por push para o repositório onde você o configurou. Você verá uma resposta da função do Lambda na guia Monitoramento da função no console do Lambda. Na guia Monitoramento, escolha Exibir logs CloudWatch. O CloudWatch console abre em uma nova guia e exibe os eventos da sua função. Selecione o fluxo de log na lista que corresponde à hora em que você enviou a sua confirmação por push. Você deve ver dados de evento semelhantes a:

```
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
```

Exemplo: criar um gatilho AWS CodeCommit para uma AWS Lambda função existente

A forma mais fácil de criar um gatilho que invoque uma função do Lambda é criá-lo no console do Lambda. Essa integração integrada garante que CodeCommit ele tenha as permissões necessárias para executar a função. Para adicionar um gatilho a uma função do Lambda, acesse o console do Lambda e escolha a função. Na guia gatilhos (Gatilhos) da função, siga as etapas descritas em [Add gatilho \(Adicionar gatilho\)](#). Essas etapas são semelhantes às mostradas em [Criar a função do Lambda](#).

Você também pode criar um gatilho para uma função Lambda em um CodeCommit repositório. Isso requer que você escolha uma função do lambda existente para invocar. Também requer que você configure manualmente as permissões necessárias CodeCommit para executar a função.

Tópicos

- [Configure manualmente as permissões para permitir CodeCommit a execução de uma função Lambda](#)
- [Crie um gatilho para a função Lambda em um CodeCommit repositório \(console\)](#)
- [Crie um gatilho para uma função Lambda para um CodeCommit repositório \(\)AWS CLI](#)

Configure manualmente as permissões para permitir CodeCommit a execução de uma função Lambda

Se você criar um gatilho CodeCommit que invoque uma função Lambda, deverá configurar manualmente as permissões que permitem CodeCommit executar a função Lambda. Para evitar a configuração manual, considere criar o gatilho para a função no console do Lambda.

Para permitir CodeCommit a execução de uma função Lambda

1. Abra um editor de texto simples e crie um arquivo JSON que especifique o nome da função Lambda, os detalhes do CodeCommit repositório e as ações que você deseja permitir no Lambda, semelhante ao seguinte:

```
{
  "FunctionName": "MyCodeCommitFunction",
  "StatementId": "1",
  "Action": "lambda:InvokeFunction",
  "Principal": "codecommit.amazonaws.com",
  "SourceArn": "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo",
  "SourceAccount": "111122223333"
}
```

2. Salve o arquivo como um arquivo JSON com um nome fácil de lembrar (por exemplo, *AllowAccessfromMyDemoRepo.json*).
3. Com o arquivo JSON que acabou de criar, no terminal (Linux, macOS ou Unix) ou na linha de comando (Windows), execute o comando `aws lambda add-permission` para adicionar uma permissão à política de recursos associada à função do Lambda:

```
aws lambda add-permission --cli-input-json file://AllowAccessfromMyDemoRepo.json
```

Esse comando retorna o JSON da declaração de política que você acabou de adicionar, semelhante ao seguinte:

```
{
  "Statement": "{ \"Condition\": { \"StringEquals\": { \"AWS:SourceAccount\": \"111122223333\" }, \"ArnLike\": { \"AWS:SourceArn\": \"arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo\" } }, \"Action\": [ \"lambda:InvokeFunction\" ], \"Resource\": \"arn:aws:lambda:us-east-1:111122223333:function:MyCodeCommitFunction\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"codecommit.amazonaws.com\" }, \"Sid\": \"1\" }"
```

```
}
```

Para obter mais informações sobre políticas de recursos para funções Lambda, consulte [AddPermissionOs modelos de Pull/Push eventos no Guia](#) do AWS Lambda usuário.

4. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
5. No painel de navegação do Painel, escolha Funções e, na lista de funções, selecione *lambda_basic_execution*.
6. Na página de resumo da função, escolha a guia Permissions (Permissões) e, em Inline Policies (Políticas em linha), selecione Create Role Policy (Criar política de função).
7. Na página Set Permissions, selecione Policy Generator e Select.
8. Na página Editar permissões, faça o seguinte:
 - Em Effect, selecione Allow.
 - Em Serviço da AWS , selecione AWS CodeCommit.
 - Em Ações, selecione GetRepository.
 - Em Amazon Resource Name (ARN) (Nome de recurso da Amazon (ARN)), insira o ARN do repositório (por exemplo, `arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo`).

Escolha Add Statement (Adicionar instrução) e Next Step (Próxima etapa).

9. Na página Review Policy, selecione Apply Policy.

O relatório de políticas será semelhante ao seguinte exemplo:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt11111111",
      "Effect": "Allow",
      "Action": [
        "codecommit:GetRepository"
      ],

```

```
    "Resource": [
      "arn:aws:codecommit:us-east-1:111122223333:MyDemoRepo"
    ]
  }
}
```

Crie um gatilho para a função Lambda em um CodeCommit repositório (console)

Depois de criar a função Lambda, você pode criar um gatilho CodeCommit que executa a função em resposta aos eventos do repositório que você especificar.

Note

Antes de testar ou executar com êxito o gatilho do exemplo, você deve configurar as políticas que permitem CodeCommit invocar a função e a função Lambda para obter informações sobre o repositório. Para obter mais informações, consulte [Para permitir CodeCommit a execução de uma função Lambda](#).

Para criar um gatilho para uma função do Lambda

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja criar gatilhos de eventos de repositório.
3. No painel de navegação do repositórios, selecione Settings (Configurações) e gatilhos (Gatilhos).
4. Escolha Criar acionador.
5. Em Create gatilho (Criar gatilho), faça o seguinte:
 - Em Nome do acionador, insira um nome para o gatilho (por exemplo, *MyLambdaFunctionTrigger*).
 - Em Eventos, escolha os eventos de repositório que acionarão a função do Lambda.

Se você selecionar All repository events, não poderá escolher nenhum outro evento. Se desejar escolher um subconjunto de eventos, limpe All repository events e, em seguida, escolha os eventos que deseja na lista. Por exemplo, se você desejar que o gatilho seja

executado somente quando um usuário criar uma ramificação ou uma tag no repositório do CodeCommit remova All repository events (Todos os eventos de repositório) e selecione Create branch or tag (Criar ramificação ou tag).

- Se você deseja que o gatilho seja aplicado a todas as ramificações do repositório, em Branches (Ramificações), deixe a seleção em branco, pois essa opção padrão aplica o gatilho a todas as ramificações automaticamente. Se você quiser que esse gatilho seja aplicado somente a ramificações específicas, escolha até 10 nomes de ramificação na lista de ramificações do repositório.
 - Em Choose the service to use (Escolher o serviço para usar), selecione AWS Lambda.
 - Na função do Lambda, escolha o nome da função na lista ou insira o ARN da função.
 - (Opcional) Em Dados personalizados, insira as informações que deseja incluir na função do Lambda (por exemplo, o nome de um canal IRC que os desenvolvedores usam quando debatem o desenvolvimento no repositório). Este campo é uma string. Ele não pode ser usado para passar nenhum parâmetro dinâmico.
6. (Opcional) Selecione Test gatilho (Testar gatilho). Essa opção tenta chamar a função com os dados de exemplo sobre o seu repositório, incluindo o ID de confirmação mais recente dele. (Se não houver histórico de confirmações, serão gerados valores de exemplo que consistem em zeros.) Isso ajuda você a confirmar que você configurou corretamente o acesso entre CodeCommit e a função Lambda.
 7. Selecione Create gatilho (Criar gatilho) para concluir a criação do gatilho.
 8. Para verificar a funcionalidade do gatilho, crie e envie uma confirmação por push para o repositório onde você o configurou. Você verá uma resposta da função do Lambda na guia Monitoramento da função no console do Lambda.

Crie um gatilho para uma função Lambda para um CodeCommit repositório ()AWS CLI

Você também pode usar a linha de comando para criar um gatilho para uma função Lambda em resposta a eventos do CodeCommit repositório, como quando alguém envia um commit para seu repositório.

Para criar um gatilho para uma função do Lambda

1. Abra um editor de texto simples e crie um arquivo JSON que especifique:
 - O nome de função do Lambda.

- O repositório e as ramificações que deseja monitorar com o gatilho. (Se você não especificar nenhuma ramificação, o gatilho se aplicará a todas as ramificações no repositório.)
- Os eventos que ativam o gatilho.

Salve o arquivo.

Por exemplo, se você quiser criar um gatilho para um repositório chamado *MyDemoRepo* que publique todos os eventos do repositório em uma função Lambda nomeada *MyCodeCommitFunction* para duas ramificações, e: *main* *preprod*

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "name": "MyLambdaFunctionTrigger",
      "destinationArn": "arn:aws:lambda:us-east-1:111122223333:function:MyCodeCommitFunction",
      "customData": "",
      "branches": [
        "main", "preprod"
      ],
      "events": [
        "all"
      ]
    }
  ]
}
```

Deve haver um bloco do gatilho no JSON para cada gatilho de um repositório. Para criar mais de um gatilho para um repositório, inclua blocos adicionais no JSON. Lembre-se de que todos os gatilhos criados no arquivo são para o repositório especificado. Você não pode criar gatilhos para vários repositórios em um único arquivo JSON. Por exemplo, se você desejasse criar dois gatilhos para um repositório, poderia criar um arquivo JSON com dois blocos de gatilhos. No exemplo a seguir, não há ramificação especificada no segundo bloco de gatilhos, então ele se aplica a todas as ramificações:

```
{
  "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"
      ]
    }
  ]
}

```

É possível criar gatilhos para eventos que você especifica, como quando uma confirmação é enviada por push para um repositório. Os tipos de eventos incluem:

- `all` para todos os eventos no repositório e nas ramificações especificados.
- `updateReference` para quando confirmações são enviadas por push para o repositório e a ramificações especificados.
- `createReference` para quando uma nova ramificação ou tag é criada no repositório especificado.
- `deleteReference` para quando uma nova ramificação ou tag é excluída no repositório especificado.

 Note

É possível usar mais de um tipo de evento em um gatilho. Contudo, se você especificar `all`, não poderá especificar outros eventos.

Para ver a lista completa dos tipos de eventos válidos em um terminal ou prompt de comando, insira `aws codecommit put-repository-triggers help`.

Além disso, é possível incluir uma string em `customData` (por exemplo, o nome de um canal IRC que os desenvolvedores usam quando debatem o desenvolvimento neste repositório). Este campo é uma string. Ele não pode ser usado para passar nenhum parâmetro dinâmico. Essa string é anexada como um atributo ao CodeCommit JSON retornado em resposta ao acionador.

2. (Opcional) No terminal ou no aviso de comando, execute o comando `test-repository-triggers`. Por exemplo, o seguinte é usado para testar se o arquivo JSON nomeado `trigger.json` é válido e CodeCommit pode acionar a função Lambda. Esse teste usa dados de amostra para funcionar se não houver dados reais disponíveis.

```
aws codecommit test-repository-triggers --cli-input-json file://trigger.json
```

Se houver êxito, o comando retornará informações semelhantes às seguintes:

```
{
  "successfulExecutions": [
    "MyLambdaFunctionTrigger"
  ],
  "failedExecutions": []
}
```

3. Em um terminal ou prompt de comando, execute o comando `put-repository-triggers` para criar o gatilho no CodeCommit. Por exemplo, para usar um arquivo JSON chamado `trigger.json` para criar o acionador:

```
aws codecommit put-repository-triggers --cli-input-json
file://trigger.json
```

Esse comando retorna um ID de configuração semelhante ao seguinte:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

4. Para visualizar a configuração do gatilho, execute o comando `get-repository-triggers`, especificando o nome do repositório:

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo
```

Esse comando retorna a estrutura de todos os gatilhos configurados para o repositório semelhante à seguinte::

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE",
  "triggers": [
    {
      "events": [
        "all"
      ],
      "destinationArn": "arn:aws:lambda:us-east-1:111122223333:MyCodeCommitFunction",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyLambdaFunctionTrigger",
      "customData": "Project ID 12345"
    }
  ]
}
```

5. Para testar a funcionalidade do gatilho, crie e envie uma confirmação por push para o repositório onde você o configurou. Você verá uma resposta da função do Lambda na guia Monitoramento da função no console do Lambda.

Editar gatilhos para um repositório AWS CodeCommit

Você pode editar os acionadores que foram criados para um CodeCommit repositório. Você pode alterar os eventos e as ramificações do gatilho, a ação tomada em resposta ao evento e outras configurações.

Tópicos

- [Editar um gatilho para um repositório \(console\)](#)
- [Editar um gatilho para um repositório \(AWS CLI\)](#)

Editar um gatilho para um repositório (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja editar gatilhos de eventos de repositório.
3. No painel de navegação do repositórios, selecione Settings (Configurações) e gatilhos (Gatilhos).
4. Na lista de gatilhos do repositório, selecione o gatilho que deseja editar e selecione Edit (Editar).
5. Faça as alterações desejadas no gatilho e selecione Save (Salvar).

Editar um gatilho para um repositório (AWS CLI)

1. Em um terminal (Linux, macOS ou Unix) ou prompt de comando (Windows), execute o comando `get-repository-triggers` para criar um arquivo JSON com a estrutura de todos os gatilhos configurados para o seu repositório. Por exemplo, para criar um arquivo JSON chamado *MyTriggers.json* com a estrutura de todos os acionadores configurados para um repositório chamado: *MyDemoRepo*

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>MyTriggers.json
```

Esse comando não retorna nada, mas um arquivo chamado *MyTriggers.json* é criado no diretório em que você executou o comando.

2. Edite o arquivo JSON em um editor de texto plano e faça as alterações no bloco de gatilhos do gatilho que deseja editar. Substitua o par `configurationId` por um par `repositoryName`. Salve o arquivo.

Por exemplo, se você quiser editar um gatilho chamado *MyFirstTrigger* no repositório nomeado para *MyDemoRepo* que ele se aplique a todas as ramificações `repositoryName`, `configurationId` substitua por e remova o especificado `main` e as `preprod` ramificações

em *red italic text*. Por padrão, se nenhuma ramificação for especificada, o gatilho será aplicado a todas as ramificações no repositório:

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    }
  ]
}
```

3. No terminal ou na linha de comando, execute o comando `put-repository-triggers`. Isso atualiza todos os acionadores do repositório, incluindo as alterações que você fez no gatilho: *MyFirstTrigger*

```
aws codecommit put-repository-triggers --repository-name MyDemoRepo
file:///MyTriggers.json
```

Esse comando retorna um ID de configuração semelhante ao seguinte:

```
{
  "configurationId": "0123456-I-AM-AN-EXAMPLE"
}
```

Acionadores de teste para um repositório AWS CodeCommit

Você pode testar os acionadores que foram criados para um CodeCommit repositório. O teste envolve executar o trigger com dados de amostra do seu repositório, incluindo o ID de confirmação mais recente. Se não existir nenhum histórico de confirmação para o repositório, em vez disso serão

gerados valores de exemplo constituídos por zeros. Testar gatilhos ajuda você a confirmar que você configurou corretamente o acesso entre CodeCommit e o destino do gatilho, seja uma AWS Lambda função ou uma notificação do Amazon Simple Notification Service.

Tópicos

- [Testar um gatilho para um repositório \(console\)](#)
- [Testar um gatilho para um repositório \(AWS CLI\)](#)

Testar um gatilho para um repositório (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja testar triggers de eventos de repositório.
3. No painel de navegação do repositórios, selecione Settings (Configurações) e gatilhos (Gatilhos).
4. Escolha o gatilho que você deseja testar e selecione Test trigger (Testar gatilho). Você deverá ver uma mensagem de êxito ou falha. Se obtiver êxito, você deverá visualizar uma resposta de ação correspondente da função do Lambda ou do tópico do Amazon SNS.

Testar um gatilho para um repositório (AWS CLI)

1. Em um terminal (Linux, macOS ou Unix) ou prompt de comando (Windows), execute o comando `get-repository-triggers` para criar um arquivo JSON com a estrutura de todos os gatilhos configurados para o seu repositório. Por exemplo, para criar um arquivo JSON chamado *TestTrigger.json* com a estrutura de todos os acionadores configurados para um repositório chamado: MyDemoRepo

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>TestTrigger.json
```

Esse comando cria um arquivo nomeado *TestTriggers.json* no diretório em que você executou o comando.

2. Edite o arquivo JSON em um editor de texto plano e faça as alterações na declaração do trigger. Substitua o par `configurationId` por um par `repositoryName`. Salve o arquivo.

Por exemplo, se você quiser testar um gatilho chamado *MyFirstTrigger* no repositório nomeado para *MyDemoRepo* que ele se aplique a todas as ramificações, `configurationId` substitua o por `repositoryName` e salve um arquivo semelhante ao seguinte, como *TestTrigger.json*:

```
{
  "repositoryName": "MyDemoRepo",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-east-2:111122223333:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    }
  ]
}
```

3. No terminal ou na linha de comando, execute o comando `test-repository-triggers`. Isso atualiza todos os acionadores do repositório, incluindo as alterações que você fez no gatilho: *MyFirstTrigger*

```
aws codecommit test-repository-triggers --cli-input-json file://TestTrigger.json
```

Esse comando retorna uma resposta semelhante à seguinte:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

Excluir gatilhos de um repositório AWS CodeCommit

Você pode excluir gatilhos que não estão mais sendo usados. Não é possível desfazer a exclusão de um gatilho, mas é possível recriar um novamente.

Note

Se você tiver configurado gatilhos para seu repositório, a exclusão do repositório não exclui os tópicos do Amazon SNS nem as funções do Lambda que foram configurados como destinos desses gatilhos. Não se esqueça de excluir esses recursos também, se não forem mais necessários.

Tópicos

- [Excluir um gatilho de um repositório \(console\)](#)
- [Excluir um gatilho de um repositório \(AWS CLI\)](#)

Excluir um gatilho de um repositório (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja excluir gatilhos de eventos de repositório.
3. No painel de navegação do repositório, escolha Settings. Em Settings, escolha gatilhos.
4. Selecione o gatilho que deseja excluir na lista de gatilhos e selecione Delete (Excluir).
5. Na caixa de diálogo, digite delete (excluir) para confirmar.

Excluir um gatilho de um repositório (AWS CLI)

1. Em um terminal (Linux, macOS ou Unix) ou prompt de comando (Windows), execute o comando `get-repository-triggers` para criar um arquivo JSON com a estrutura de todos os gatilhos configurados para o seu repositório. Por exemplo, para criar um arquivo JSON chamado *MyTriggers.json* com a estrutura de todos os acionadores configurados para um repositório chamado: MyDemoRepo

```
aws codecommit get-repository-triggers --repository-name MyDemoRepo  
>MyTriggers.json
```

Esse comando cria um arquivo nomeado *MyTriggers.json* no diretório em que você executou o comando.

2. Edite o arquivo JSON em um editor de texto simples e remova o bloco de gatilho do gatilho que você deseja excluir. Substitua o par `configurationId` por um par `repositoryName`. Salve o arquivo.

Por exemplo, se você quiser remover um gatilho chamado *MyFirstTrigger* do repositório chamado *MyDemoRepo* `repositoryName`, `configurationId` substitua por e remova a instrução em *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. No terminal ou na linha de comando, execute o comando `put-repository-triggers`. Isso atualiza os acionadores do repositório e exclui o gatilho: *MyFirstTrigger*


```
aws codecommit put-repository-triggers --repository-name MyDemoRepo  
file://MyTriggers.json
```

Esse comando retorna um ID de configuração semelhante ao seguinte:

```
{  
  "configurationId": "0123456-I-AM-AN-EXAMPLE"  
}
```

Note

Para excluir todos os gatilhos de um repositório chamado *MyDemoRepo*, o arquivo JSON seria semelhante a:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": []  
}
```

Associar ou desassociar um AWS CodeCommit repositório com o Amazon Reviewer CodeGuru

O Amazon CodeGuru Reviewer é um serviço automatizado de revisão de código que usa análise de programas e aprendizado de máquina para detectar problemas comuns e recomendar correções em seu código Java ou Python. Você pode associar repositórios na sua conta da Amazon Web Services ao CodeGuru Reviewer. Ao fazer isso, o CodeGuru revisor cria uma função vinculada ao serviço que permite ao CodeGuru revisor analisar o código em todas as pull requests criadas após a associação ser feita.

Depois de associar um repositório, o CodeGuru revisor analisa e comenta os problemas encontrados ao criar pull requests. Cada comentário está claramente marcado como proveniente de um CodeGuru revisor com a designação Amazon CodeGuru Reviewer. É possível responder a esses comentários exatamente como qualquer outro comentário em uma solicitação pull e também fornecer feedback sobre a qualidade da sugestão. Esse feedback é compartilhado com o CodeGuru revisor e pode ajudar a melhorar o serviço e suas sugestões.

 Note

Você não verá comentários do CodeGuru Reviewer em pull requests que foram criados antes de o repositório ser associado a ele. Talvez você não veja comentários em solicitações pull criadas após a associação pelos seguintes motivos:

- A solicitação pull não contém códigos Java ou Python.
- CodeGuru O revisor não teve tempo suficiente para executar e revisar o código na pull request. Esse processo pode demorar até 30 minutos. Os comentários podem aparecer à medida que a revisão avança, mas os comentários não são concluídos até que o status do trabalho seja exibido como Concluído.
- CodeGuru O revisor não encontrou nenhum problema no código Java ou Python na pull request.
- Ocorreu uma falha ao executar o trabalho de revisão de código. Para revisar o status de uma revisão de uma solicitação pull, consulte a guia Atividade da solicitação pull.
- Você está visualizando as alterações na pull request na guia Changes, a pull request foi atualizada e o Amazon CodeGuru Reviewer não encontrou nenhum problema nas alterações. Os comentários do Amazon CodeGuru Reviewer só aparecem na guia Alterações se tiverem sido feitos na revisão mais recente do pull request. Eles sempre aparecem na guia Atividade.

The screenshot shows the AWS CodeCommit pull request interface. At the top, the breadcrumb navigation is 'Developer Tools > CodeCommit > Repositories > MyDemoRepo > Pull requests > 25'. The title of the pull request is '25: Updated some of our Java samples'. On the right, there are buttons for 'Approve', 'Close pull request', and 'Merge'. Below the title, there are tabs for 'Open', 'No approval rules', 'No merge conflicts', 'Destination main', 'Source bugfix-1236', 'Author: Li_Juan', and 'Approvals: 0'. There are also tabs for 'Details', 'Activity', 'Changes', 'Commits', and 'Approvals'. The 'Activity' tab is selected. Under 'Activity', there is a section for 'Amazon CodeGuru Reviewer job status' with a status of 'In progress'. Below that, there is an 'Activity history' section with a blue bar indicating 'Pull request updated 1 minute ago. One or more commits added. Li_Juan updated the pull request.' Below the activity history, there is a comment on line 100 of 'EventHandler.java' from 'Amazon CodeGuru Reviewer' (Bot) 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.' There is a 'Reply' button and a thumbs up icon with a count of 1.

Para obter mais informações, consulte [Trabalhando com pull requests em AWS CodeCommit repositórios](#) e [Revisar uma solicitação pull](#), e o [Guia do usuário do Amazon CodeGuru Reviewer](#).

Note

Você deve estar conectado com um usuário ou função do IAM que tenha permissões suficientes para associar ou desassociar um repositório ao Reviewer. CodeGuru Para obter informações sobre as políticas gerenciadas CodeCommit que incluem essas permissões, consulte [AWS políticas gerenciadas para CodeCommit](#) [AWS CodeCommit políticas gerenciadas](#) e [Amazon CodeGuru Reviewer](#) e. Para obter informações sobre permissões e segurança do CodeGuru revisor, consulte o Guia do usuário do Amazon CodeGuru Reviewer.

Tópicos

- [Associar um repositório ao Reviewer CodeGuru](#)
- [Desassociar um repositório do Reviewer CodeGuru](#)

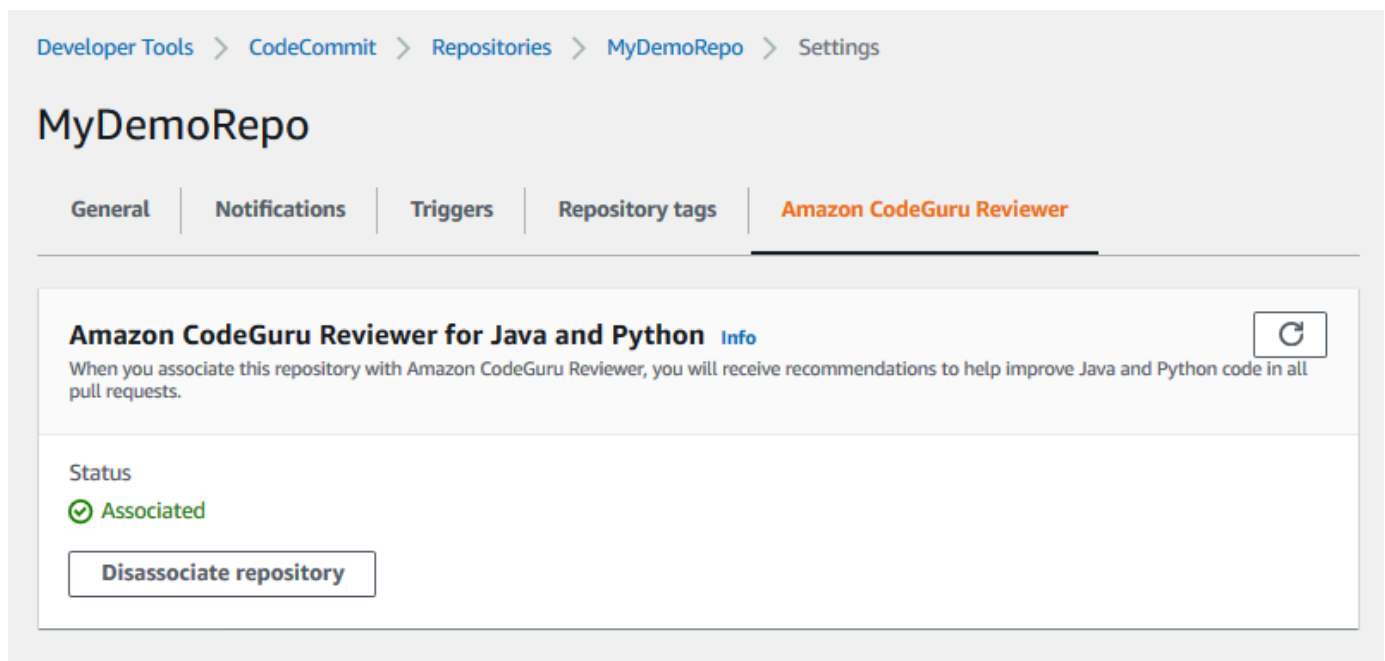
Associar um repositório ao Reviewer CodeGuru

Use o AWS CodeCommit console para associar rapidamente um repositório ao CodeGuru Reviewer. Para outros métodos, consulte o Guia do usuário do Amazon CodeGuru Reviewer.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositórios, escolha o nome do repositório a ser associado CodeGuru ao Reviewer.
3. Escolha Configurações e, em seguida, escolha Amazon CodeGuru Reviewer.
4. Escolha Associate repository (Associar repositório).

Note

Pode levar até 10 minutos para associar totalmente um repositório ao CodeGuru Reviewer. O status não será atualizado automaticamente. Para visualizar o status atual, clique no botão para atualizar.



Desassociar um repositório do Reviewer CodeGuru

Use o AWS CodeCommit console para dissociar rapidamente um repositório do CodeGuru Reviewer. Para outros métodos, consulte o Guia do usuário do Amazon CodeGuru Reviewer.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositórios, escolha o nome do repositório que você deseja desassociar do Revisor. CodeGuru
3. Escolha Configurações e, em seguida, escolha Amazon CodeGuru Reviewer.
4. Escolha Disassociate repository (Desassociar repositório).

Exibir detalhes CodeCommit do repositório

Você pode usar o AWS CodeCommit console ou o Git de um repositório local conectado ao repositório para visualizar informações sobre os CodeCommit repositórios disponíveis. AWS CLI

Antes de seguir estas instruções, conclua as etapas em [Configurando](#).

Tópicos

- [Visualizar de detalhes de repositórios \(console\)](#)
- [Exibir detalhes CodeCommit do repositório \(Git\)](#)
- [Exibir detalhes CodeCommit do repositório \(\)AWS CLI](#)

Visualizar de detalhes de repositórios (console)

Use o AWS CodeCommit console para visualizar rapidamente todos os repositórios criados com sua conta da Amazon Web Services.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositórios, veja os detalhes sobre os repositórios na Região da AWS onde você fez login. Use o seletor de regiões para escolher uma Região da AWS diferente para visualizar os repositórios nessa região.
3. Escolha o nome do repositório para o qual você deseja ver mais detalhes e realize uma das seguintes ações:
 - Para visualizar a URL para clonar o repositório, escolha Clone URL (Clonar URL) e o protocolo que deseja usar ao clonar o repositório no painel de navegação. Isso copia a URL do clone. Para revisá-la, cole-a em um editor de texto não criptografado.
 - Para visualizar as opções configuráveis do repositório, bem como detalhes como o ARN e o ID do repositório, no painel de navegação, escolha Configurações.

 Note

Se você entrou como um usuário do IAM, é possível configurar e salvar suas preferências para visualizar o código e outras configurações do console. Para obter mais informações, consulte [Trabalhar com preferências do usuário](#).

Exibir detalhes CodeCommit do repositório (Git)

Para usar o Git de um repositório local para ver detalhes sobre CodeCommit repositórios, execute o comando. `git remote show`

Antes de executar essas etapas, conecte o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

1. Execute o `git remote show remote-name` comando, onde **remote-name** está o alias do CodeCommit repositório (por padrão, `origin`).

 Tip

Para obter uma lista dos nomes dos CodeCommit repositórios e seus URLs, execute o `git remote -v` comando.

Por exemplo, para ver detalhes sobre o CodeCommit repositório com o `origin` alias:

```
git remote show origin
```

2. Para HTTPS:

```
* remote origin
Fetch URL: https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
Push URL: https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
HEAD branch: (unknown)
Remote branches:
  MyNewBranch tracked
  main tracked
Local ref configured for 'git pull':
  MyNewBranch merges with remote MyNewBranch (up to date)
Local refs configured for 'git push':
```

```
MyNewBranch pushes to MyNewBranch (up to date)
main pushes to main (up to date)
```

Para o SSH:

```
* remote origin
Fetch URL: ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
Push URL: ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
HEAD branch: (unknown)
Remote branches:
  MyNewBranch tracked
  main tracked
Local ref configured for 'git pull':
  MyNewBranch merges with remote MyNewBranch (up to date)
Local refs configured for 'git push':
  MyNewBranch pushes to MyNewBranch (up to date)
  main pushes to main (up to date)
```

 Tip

Para pesquisar o ID da chave SSH do seu usuário do IAM, abra o console do IAM e expanda Credenciais de segurança na página de detalhes do usuário do IAM. O ID da chave SSH pode ser encontrado em Chaves SSH para. AWS CodeCommit

Para obter mais opções, consulte a documentação do Git.

Exibir detalhes CodeCommit do repositório (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para visualizar os detalhes do repositório, execute os seguintes comandos:

- Para ver uma lista de nomes de CodeCommit repositórios e seus correspondentes IDs, execute [list-repositories](#).
- Para visualizar informações sobre um único CodeCommit repositório, execute [get-repository](#).
- Para visualizar informações sobre vários repositórios em CodeCommit, execute [batch-get-repositories](#).

Para ver uma lista de CodeCommit repositórios

1. Execute o comando `list-repositories`:

```
aws codecommit list-repositories
```

Você pode usar as opções `--sort-by` ou `--order` para alterar a ordem das informações retornadas.

2. Se for bem-sucedido, esse comando exibirá um `repositories` objeto que contém os nomes e IDs de todos os repositórios CodeCommit associados à conta da Amazon Web Services.

Veja um exemplo de saída com base no comando do exemplo anterior:

```
{
  "repositories": [
    {
      "repositoryName": "MyDemoRepo",
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
    },
    {
      "repositoryName": "MyOtherDemoRepo",
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
    }
  ]
}
```

Para ver detalhes sobre um único CodeCommit repositório

1. Execute o `get-repository` comando, especificando o nome do CodeCommit repositório com a `--repository-name` opção.

Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

Por exemplo, para ver detalhes sobre um CodeCommit repositório chamado `MyDemoRepo`:


```
aws codecommit get-repository --repository-name MyDemoRepo
```

2. Se houver êxito, o comando gerará como saída um objeto `repositoryMetadata` com as seguintes informações:

- O nome do repositório (`repositoryName`).
- A descrição do repositório (`repositoryDescription`).
- O ID exclusivo do repositório gerado pelo sistema (`repositoryId`).
- O ID da conta da Amazon Web Services associada ao repositório (`accountId`).

Veja um exemplo de saída, com base no comando do exemplo anterior:

```
{
  "repositoryMetadata": {
    "creationDate": 1429203623.625,
    "defaultBranch": "main",
    "repositoryName": "MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "lastModifiedDate": 1430783812.0869999,
    "repositoryDescription": "My demonstration repository",
    "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
    "accountId": "111111111111"
  }
}
```

Para ver detalhes sobre vários CodeCommit repositórios

1. Execute o comando `batch-get-repositories` com a opção `--repository-names`. Adicione um espaço entre o nome de cada CodeCommit repositório.

Tip

Para obter os nomes dos repositórios CodeCommit, execute o comando [list-repositories](#).

Por exemplo, para ver detalhes sobre dois CodeCommit repositórios chamados MyDemoRepo e MyOtherDemoRepo:

```
aws codecommit batch-get-repositories --repository-names MyDemoRepo MyOtherDemoRepo
```

2. Se houver êxito, o comando gerará como saída um objeto com as seguintes informações:

- Uma lista de todos CodeCommit os repositórios que não puderam ser encontrados (`repositoriesNotFound`).
- Uma lista de CodeCommit repositórios (`repositories`). Cada nome de CodeCommit repositório é seguido por:
 - A descrição do repositório (`repositoryDescription`).
 - O ID exclusivo do repositório gerado pelo sistema (`repositoryId`).
 - O ID da conta da Amazon Web Services associada ao repositório (`accountId`).

Veja um exemplo de saída, com base no comando do exemplo anterior:

```
{
  "repositoriesNotFound": [],
  "repositories": [
    {
      "creationDate": 1429203623.625,
      "defaultBranch": "main",
      "repositoryName": "MyDemoRepo",
      "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
      "lastModifiedDate": 1430783812.0869999,
      "repositoryDescription": "My demonstration repository",
      "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
      "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
      "accountId": "111111111111"
    },
    {
      "creationDate": 1429203623.627,
      "defaultBranch": "main",
      "repositoryName": "MyOtherDemoRepo",
```

```
        "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyOtherDemoRepo",
        "lastModifiedDate": 1430783812.0889999,
        "repositoryDescription": "My other demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/MyOtherDemoRepo",
        "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo",
        "accountId": "111111111111"
    }
],
"repositoriesNotFound": []
}
```

Alterar as AWS CodeCommit configurações do repositório

Você pode usar o console AWS CLI e o AWS CodeCommit console para alterar as configurações de um CodeCommit repositório, como sua descrição ou nome.

Important

A alteração do nome de um repositório pode corromper repositórios locais que usem o nome antigo no URL remoto. Execute o comando `git remote set-url` para atualizar a URL remota para usar o novo nome do repositório.

Tópicos

- [Alteração de configurações de repositório \(console\)](#)
- [Alterar as configurações AWS CodeCommit do repositório \(AWS CLI\)](#)

Alteração de configurações de repositório (console)

Para usar o AWS CodeCommit console para alterar as configurações de um CodeCommit repositório AWS CodeCommit, siga estas etapas.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja alterar as configurações.

3. No painel de navegação, selecione Configurações.
4. Para alterar o nome do repositório, em Repository name (Nome do repositório), insira um nome novo na caixa de texto Name (Nome) e selecione Save (Salvar). Quando solicitado, verifique sua opção.

Important

A alteração do nome do AWS CodeCommit repositório mudará o SSH e o HTTPS URLs que os usuários precisam para se conectar ao repositório. Os usuários não poderão se conectar ao repositório até que atualizem suas configurações de conexão. Além disso, devido à alteração no ARN do repositório, alterar o nome do repositório invalidará todas as políticas de usuário do IAM que dependem do ARN desse repositório.

Para se conectar ao repositório após o nome ter sido alterado, cada usuário deve usar o comando `git remote set-url` e especificar a nova URL a ser usada. Por exemplo, se você alterou o nome do repositório de `MyDemoRepo` para `MyRenamedDemoRepo`, os usuários que usam HTTPS para se conectar ao repositório executarão o seguinte comando Git:

```
git remote set-url origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Os usuários usando o SSH para se conectar ao repositório executariam o seguinte comando do Git:

```
git remote set-url origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyRenamedDemoRepo
```

Para obter mais opções, consulte a documentação do Git.

5. Para alterar a descrição do repositório, modifique o texto na caixa de texto Description (Descrição) e selecione Save (Salvar).

Note

O campo de descrição exibe o Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o `GetRepository` ou `BatchGetRepositories` APIs e planeja exibir o

campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

6. Para alterar a ramificação padrão, na Default branch (Ramificação padrão), escolha a lista suspensa da ramificação e escolha uma ramificação diferente. Escolha Salvar.
7. Para alterar a chave de AWS KMS criptografia usada para criptografar e descriptografar dados no repositório, em Chave de criptografia do repositório, escolha uma Chave gerenciada pela AWS ou Chave gerenciada pelo cliente para especificar o tipo de chave a ser usada. Se escolher uma chave gerenciada pelo cliente, insira o ARN da chave. Escolha Salvar.
8. Para excluir o repositório, selecione Delete repository. Na caixa ao lado de Type the name of the repository to confirm deletion (Digite o nome do repositório para confirmar a exclusão), insira **delete** e escolha Delete (Excluir).

 Important

Depois de excluir esse repositório AWS CodeCommit, você não poderá mais cloná-lo em nenhum repositório local ou repositório compartilhado. Também não será mais possível extrair dados dele nem enviar via push dados para ele de nenhum repositório local ou compartilhado. Esta ação não pode ser desfeita.

Alterar as configurações AWS CodeCommit do repositório (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar AWS CLI para alterar as configurações de um CodeCommit repositório AWS CodeCommit, execute um ou mais dos seguintes comandos:

- [update-repository-description](#) para alterar a descrição de um CodeCommit repositório.
- [update-repository-name](#) para alterar o nome de um CodeCommit repositório.

Para alterar a descrição de um CodeCommit repositório

1. Execute o comando `update-repository-description`, especificando:
 - O nome do CodeCommit repositório (com a `--repository-name` opção).

 Tip

Para obter o nome do CodeCommit repositório, execute o [list-repositories](#) comando.

- A nova descrição do repositório (com a opção `--repository-description`).

 Note

O campo de descrição exibe Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o `GetRepository` ou `BatchGetRepositories` APIs e planeja exibir o campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

Por exemplo, para alterar a descrição do CodeCommit repositório chamado `MyDemoRepo` para `This description was changed`:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --repository-description "This description was changed"
```

Esse comando gerará saídas somente se houver erros.

2. Para verificar a descrição alterada, execute o `get-repository` comando, especificando o nome do CodeCommit repositório cuja descrição você alterou com a `--repository-name` opção.

A saída do comando exibe o texto alterado em `repositoryDescription`.

Para alterar o nome de um CodeCommit repositório

1. Execute o comando `update-repository-name`, especificando:

- O nome atual do CodeCommit repositório (com a `--old-name` opção).

 Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

- O novo nome do CodeCommit repositório (com a `--new-name` opção).

Por exemplo, para alterar o repositório chamado MyDemoRepo para MyRenamedDemoRepo:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-name
MyRenamedDemoRepo
```

Esse comando gerará saídas somente se houver erros.

Important

A alteração do nome do AWS CodeCommit repositório altera o SSH e o HTTPS URLs que os usuários precisam para se conectar ao repositório. Os usuários não poderão se conectar a este repositório até que atualizem suas configurações de conexão. Além disso, devido às alterações no ARN do repositório, a alteração do nome do repositório invalidará todas as políticas de usuário do IAM que dependem do ARN desse repositório.

2. Para verificar o nome alterado, execute o comando `list-repositories` e revise a lista de nomes de repositório.

Sincronizar alterações entre um repositório local e um repositório AWS CodeCommit

Você usa o Git para sincronizar as alterações entre um repositório local e o CodeCommit repositório conectado ao repositório local.

Para enviar as alterações do repositório local para o CodeCommit repositório, execute. `git push remote-name branch-name`

Para extrair as alterações no repositório local do CodeCommit repositório, execute. `git pull remote-name branch-name`

Tanto para enviar quanto para puxar, **remote-name** é o apelido que o repositório local usa para o repositório. CodeCommit **branch-name** é o nome da ramificação no CodeCommit repositório a ser enviada ou extraída.

 Tip

Para obter o apelido que o repositório local usa para o CodeCommit repositório, execute. `git remote` Para obter uma lista de nomes de ramificação, execute `git branch`. Um asterisco (*) aparece próximo ao nome da ramificação atual. (Você também pode executar `git status` para mostrar o nome da ramificação atual.)

 Note

Se você clonou o repositório, do ponto de vista do repositório local, não *remote-name* é o nome do repositório. CodeCommit Quando você clona um repositório, *remote-name* é definido automaticamente como. `origin`

Por exemplo, para enviar alterações do repositório local para a `main` ramificação no CodeCommit repositório com o apelido: `origin`

```
git push origin main
```

Da mesma forma, para extrair alterações no repositório local da `main` ramificação no CodeCommit repositório com o apelido: `origin`

```
git pull origin main
```

 Tip

Se você adicionar a opção `-u` ao `git push`, definirá informações de rastreamento upstream. Por exemplo, se você corrigir `git push -u origin main`), no futuro poderá correr `git push` e `git pull` sem *remote-namebranch-name*. Para obter as informações de rastreamento upstream, execute `git remote show remote-name` (por exemplo, `git remote show origin`).

Para obter mais opções, consulte a documentação do Git.

Confirmações de push em um repositório do Git adicional

Você pode configurar seu repositório local para enviar alterações por push a dois repositórios remotos. Por exemplo, talvez você precise continuar usando sua solução existente de repositório do Git enquanto experimenta o AWS CodeCommit. Siga estas etapas básicas para enviar alterações em seu repositório local CodeCommit e em um repositório Git separado.

Tip

Se você não tiver um repositório Git, poderá criar um vazio em um serviço diferente CodeCommit e depois migrar seu CodeCommit repositório para ele. Você deve seguir etapas semelhantes às descritas em [Migrar para o CodeCommit](#).

1. No terminal ou no prompt de comando, mude para o diretório do repositório local e execute o comando `git remote -v`. Você deve ver uma saída semelhante a:

Para HTTPS:

```
origin  https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin  https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para o 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. Execute o `git remote set-url --add --push origin git-repository-name` comando onde **git-repository-name** está a URL e o nome do repositório Git em que você deseja hospedar seu código. Isso altera o destino do push de `origin` para esse repositório Git.

Note

`git remote set-url --add --push` substitui o URL padrão para pushes, por isso, você deve executar esse comando duas vezes, como será demonstrado em etapas posteriores.

Por exemplo, o comando a seguir altera o push of origin para *some-URL*/MyDestinationRepo:

```
git remote set-url --add --push origin some-URL/MyDestinationRepo
```

Esse comando não retorna nada.

 Tip

Se você estiver enviando para um repositório Git que exige credenciais, certifique-se de configurar essas credenciais em um auxiliar de credenciais ou na configuração da string. *some-URL* Caso contrário, as operações de envio para esse repositório falharão.

3. Execute o comando `git remote -v` novamente, o que deve criar uma saída semelhante a:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin some-URL/MyDestinationRepo (push)
```

Para o SSH:

```
origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin some-URL/MyDestinationRepo (push)
```

4. Agora adicione o CodeCommit repositório. Execute `git remote set-url --add --push origin` novamente, desta vez com a URL e o nome de repositório de seu repositório CodeCommit.

Por exemplo, o comando a seguir adiciona o push of origin a `https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo`:

Para HTTPS:

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

Para o SSH:

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

Esse comando não retorna nada.

5. Execute o comando `git remote -v` novamente, o que deve criar uma saída semelhante a:

Para HTTPS:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin some-URL/MyDestinationRepo (push)
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Para o 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)
```

Agora você tem dois repositórios Git como destino para seus push, mas seus push vão para / primeiro. *some-URL* MyDestinationRepo Se o push ao repositório falhar, suas confirmações não serão enviadas por push para nenhum dos repositórios.

Tip

Se o outro repositório exigir credenciais que você queira inserir manualmente, considere alterar a ordem dos envios para que você os envie primeiro. CodeCommit Execute `git remote set-url --delete` para excluir o repositório para onde o push foi enviado primeiro e, depois, execute `git remote set-url --add` para adicioná-lo novamente para que se torne o segundo destino do push na lista.

Para obter mais opções, consulte a documentação do Git.

6. Para verificar se você agora está enviando por push para ambos os repositórios remotos, use um editor de texto para criar o seguinte arquivo de texto no repositório local:

```
bees.txt
-----
```

```
Bees are flying insects closely related to wasps and ants, and are known for their
role in pollination and for producing honey and beeswax.
```

7. Execute git add para preparar a alteração no repositório local:

```
git add bees.txt
```

8. Execute git commit para confirmar a alteração no repositório local:

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

9. Para enviar o commit do repositório local para seus repositórios remotos, execute git push -u **remote-name branch-name** where **remote-name** é o apelido que o repositório local usa para os repositórios remotos e **branch-name** é o nome da ramificação a ser enviada para o repositório.

 Tip

Você só precisa usar a opção -u na primeira vez em que você enviar por push. Em seguida, as informações de rastreamento upstream são definidas.

Por exemplo, executar git push -u origin main mostraria que o push foi enviado aos dois repositórios remotos nas ramificações esperadas, com uma saída semelhante a:

Para HTTPS:

```
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To some-URL/MyDestinationRepo
   a5ba4ed..250f6c3  main -> main
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

```
a5ba4ed..250f6c3  main -> main
```

Para o SSH:

```
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To some-URL/MyDestinationRepo
  a5ba4ed..250f6c3  main -> main
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.61 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote:
To ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
  a5ba4ed..250f6c3  main -> main
```

Para obter mais opções, consulte a documentação do Git.

Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções

Você pode configurar o acesso aos CodeCommit repositórios para usuários e grupos do IAM em outra AWS conta. Esse processo é denominado acesso entre várias contas. Esta seção fornece exemplos e step-by-step instruções para configurar o acesso entre contas a um repositório nomeado *MySharedDemoRepo* na região Leste dos EUA (Ohio) em uma AWS conta (chamada de AccountA) para usuários do IAM que pertencem a um grupo do IAM nomeado *DevelopersWithCrossAccountRepositoryAccess* em outra AWS conta (chamada de AccountB).

Esta seção é dividida em três partes:

- Ações para o administrador na AccountA.
- Ações para o administrador na AccountB.
- Ações para o usuário do repositório na AccountB.

Para configurar o acesso entre várias contas:

- O administrador em AccountA faz login como usuário do IAM com as permissões necessárias para criar e gerenciar repositórios CodeCommit e criar funções no IAM. Se você estiver usando políticas gerenciadas, aplique IAMFull Access e AWSCode CommitFullAccess a esse usuário do IAM.

O ID da conta de exemplo da AccountA é **111122223333**.

- O administrador da AccountB faz login como um usuário do IAM com as permissões necessárias para criar e gerenciar grupos e usuários do IAM e para configurar políticas para usuários e grupos. Se você estiver usando políticas gerenciadas, aplique o IAMFull Access a esse usuário do IAM.

O ID da conta de exemplo da AccountB é **888888888888**.

- O usuário do repositório no AccountB, para emular as atividades de um desenvolvedor, faz login como um usuário do IAM que é membro do grupo do IAM criado para permitir o acesso ao CodeCommit repositório no AccountA. Essa conta deve ser configurada com:
 - AWS Acesso ao console de gerenciamento.
 - Uma chave de acesso e uma chave secreta para usar ao se conectar aos AWS recursos e o ARN da função a ser assumida ao acessar repositórios na AccountA.
 - O utilitário git-remote-codecommit no computador local onde o repositório é clonado. Esse utilitário requer o Python e seu instalador, pip. É possível baixar o utilitário a partir de [git-remote-codecommit](#) no site Python Package Index.

Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#) e [Usuários do IAM](#).

Tópicos

- [Acesso ao repositório entre contas: ações para o administrador na AccountA](#)
- [Acesso ao repositório entre contas: ações para o administrador na AccountB](#)
- [Acesso ao repositório entre contas: ações para o usuário do repositório na AccountB](#)

Acesso ao repositório entre contas: ações para o administrador na AccountA

Para permitir que os usuários ou grupos na AccountB acessem um repositório na AccountA, o administrador da AccountA deve:

- Criar uma política na AccountA que concede acesso ao repositório.
- Criar uma função na AccountA que possa ser assumida por grupos e usuários do IAM na AccountB.
- Anexe a política ao perfil.

As seções a seguir apresentam etapas e exemplos.

Tópicos

- [Etapa 1: criar uma política para acesso ao repositório na AccountA](#)
- [Etapa 2: criar uma função para acesso ao repositório na AccountA](#)

Etapa 1: criar uma política para acesso ao repositório na AccountA

Você pode criar uma política na AccountA que conceda acesso ao repositório na AccountA aos usuários na AccountB. Dependendo do nível de acesso que deseja conceder, execute uma das seguintes ações:

- Configure a política para permitir que os usuários da AccountB acessem a um repositório específico, mas não permita a visualização da lista de todos os repositórios na AccountA.
- Configure o acesso adicional para permitir que os usuários da AccountB escolham o repositório em uma lista de todos os repositórios na AccountA.

Para criar uma política para acesso ao repositório

1. Faça login no AWS Management Console como um usuário do IAM com permissões para criar políticas no AccountA.
2. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
3. No painel de navegação, selecione Políticas.
4. Selecione Criar política.
5. Escolha a guia JSON e cole o documento de política JSON a seguir na caixa de texto JSON.
us-east-2 Substitua Região da AWS pelo do repositório, *111122223333* pelo ID da conta da AccountA *MySharedDemoRepo* e pelo nome do CodeCommit seu repositório na 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 você quiser que os usuários que assumem essa função possam visualizar uma lista de repositórios na página inicial do CodeCommit console, adicione uma declaração adicional à política, da seguinte forma:

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": "*"
}
]
}

```

Esse acesso facilita para os usuários que assumem essa função com essa política a localização do repositório ao qual eles têm acesso. Eles podem escolher o nome do repositório na lista e serem direcionados para a página inicial do repositório compartilhado (Code). Os usuários não podem acessar qualquer um dos outros repositórios exibidos na lista, mas podem visualizar os repositórios na AccountA na página Dashboard.

Se você não quiser permitir que os usuários que assumem a função possam visualizar uma lista de todos os repositórios no AccountA, use o primeiro exemplo de política, mas certifique-se de enviar a esses usuários um link direto para a página inicial do repositório compartilhado no console. CodeCommit

6. Selecione Revisar política. O validador de políticas reporta erros de sintaxe (por exemplo, se você esquecer de substituir o ID da conta da Amazon Web Services e o nome do repositório de exemplo pelo seu ID da conta da Amazon Web Services e nome do repositório).
7. Na página Revisar política, insira um nome para a política (por exemplo, *CrossAccountAccessForMySharedDemoRepo*). Você também pode fornecer uma descrição opcional para esta política. Selecione Criar política.

Etapa 2: criar uma função para acesso ao repositório na AccountA

Após configurar uma política, crie uma função que grupos e usuários do IAM na AccountB possam assumir e anexe a política a essa função.

Para criar uma função para acesso ao repositório

1. No console do IAM, selecione Roles (Funções).
2. Selecione Criar perfil.
3. Selecionar Outra conta da Amazon Web Services.
4. Em ID da conta, insira o ID da conta da Amazon Web Services para AccountB (por exemplo, *888888888888*). Escolha Próximo: Permissões.
5. Em Anexar políticas de permissões, selecione a política que você criou no procedimento anterior (*CrossAccountAccessForMySharedDemoRepo*). Escolha Próximo: revisar.
6. Em Nome da função, insira um nome para a função (por exemplo, *MyCrossAccountRepositoryContributorRole*). Você também pode inserir uma descrição opcional para ajudar outras pessoas a entenderem a finalidade da função.
7. Selecione Criar perfil.
8. Abra a função recém-criada e copie o ARN da função (por exemplo, *arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole*). Você precisa fornecer esse ARN para o administrador da AccountB.

Acesso ao repositório entre contas: ações para o administrador na AccountB

Para permitir que os usuários ou grupos na AccountB acessem um repositório na AccountA, o administrador da AccountB deve criar um grupo na AccountB. Esse grupo deve ser configurado com

uma política que permite que os membros do grupo assumam a função criada pelo administrador da AccountA.

As seções a seguir apresentam etapas e exemplos.

Tópicos

- [Etapa 1: criar um grupo do IAM para acesso ao repositório pelos usuários da AccountB](#)
- [Etapa 2: criar uma política e adicionar usuários ao grupo do IAM](#)

Etapa 1: criar um grupo do IAM para acesso ao repositório pelos usuários da AccountB

A maneira mais simples de gerenciar quais usuários do IAM na AccountB podem acessar o repositório na AccountA é criar um grupo do IAM na AccountB com permissão para assumir a função na AccountA e, em seguida, adicionar os usuários do IAM a esse grupo.

Para criar um grupo para acesso ao repositório entre várias contas

1. Faça login no AWS Management Console como um usuário do IAM com as permissões necessárias para criar grupos e políticas do IAM e gerenciar usuários do IAM no AccountB.
2. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
3. No console do IAM, selecione Grupos.
4. Escolha Criar novo grupo.
5. Em Nome do grupo, insira um nome para o grupo (por exemplo, *DeveloPERSWithCrossAccountRepositoryAccess*). Escolha Próxima etapa.
6. Em Attach Policy, escolha Next Step. Crie a política entre várias contas no próximo procedimento. Conclua a criação do grupo.

Etapa 2: criar uma política e adicionar usuários ao grupo do IAM

Agora que você tem um grupo, crie a política que permite aos membros desse grupo assumir a função que fornece acesso ao repositório na AccountA. Em seguida, adicione ao grupo os usuários do IAM na AccountB os quais você deseja permitir o acesso na AccountA.

Para criar uma política para o grupo e adicionar usuários a ele

1. No console do IAM, escolha Grupos e, em seguida, escolha o nome do grupo que você acabou de criar (por exemplo, *DeveloPERSWithCrossAccountRepositoryAccess*).
2. Escolha a aba Permissões. Expanda Inline Policies e escolha o link para criar uma política em linha. (Se você estiver configurando um grupo que já tem uma política em linha, escolha Create Group Policy.)
3. Escolha Custom Policy e depois Select.
4. Em Policy Name (Nome da política), digite um nome para a política (por exemplo, *AccessPolicyForSharedRepository*).
5. Em Policy Document, cole a política a seguir. Em **Resource**, substitua o ARN pelo ARN da política criada pelo administrador em AccountA (por exemplo, `arn:aws:iam:::role/`) e escolha Aplicar política. **111122223333 MyCrossAccountRepositoryContributorRole** Para obter mais informações sobre a política criada pelo administrador na AccountA, consulte [Etapa 1: criar uma política para acesso ao repositório na AccountA](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource":
        "arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole"
    }
  ]
}
```

6. Escolha a guia Users. Selecione Adicionar usuários ao grupo e, em seguida, adicione os usuários do IAM da AccountB. Por exemplo, você pode adicionar um usuário do IAM com o nome de usuário *Saanvi_Sarkar* ao grupo.

 Note

Os usuários do AccountB devem ter acesso programático, incluindo uma chave de acesso e uma chave secreta, para configurar seus computadores locais para acessar o repositório compartilhado. CodeCommit Se você estiver criando usuários do IAM, salve a

chave de acesso e a chave secreta. Para ajudar a garantir a segurança de sua conta da AWS, a chave de acesso secreta é acessível somente no momento em que é criada.

Acesso ao repositório entre contas: ações para o usuário do repositório na AccountB

Para acessar o repositório na AccountA, os usuários no grupo a AccountB devem configurar os computadores locais para acesso ao repositório. As seções a seguir apresentam etapas e exemplos.

Tópicos

- [Etapa 1: configurar o AWS CLI e o Git para que um usuário AccountB acesse o repositório no AccountA](#)
- [Etapa 2: clonar e acessar o CodeCommit repositório na AccountA](#)

Etapa 1: configurar o AWS CLI e o Git para que um usuário AccountB acesse o repositório no AccountA

Você não pode usar as chaves SSH ou as credenciais do Git para acessar repositórios em outra conta da Amazon Web Services. Os usuários do AccountB devem configurar seus computadores para usar um git-remote-codecommit (recomendado) ou o auxiliar de credenciais para acessar o repositório CodeCommit compartilhado no AccountA. No entanto, você pode continuar a usar as chaves SSH ou as credenciais do Git ao acessar repositórios na AccountB.

Siga estas etapas para configurar o acesso usando git-remote-codecommit. Se você ainda não tiver instalado o git-remote-codecommit, baixe-o de [git-remote-codecommit](#) no site Python Package Index.

Para configurar o AWS CLI e o Git para acesso entre contas

1. Instale o AWS CLI no computador local. Consulte instruções para seu sistema operacional em [Instalação da AWS CLI](#).
2. Instale o Git no computador local. Para instalar o Git, recomendamos sites como [Git Downloads](#) ou [Git for Windows](#).

 Note

CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git. O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em: [Solução de problemas](#)

3. No terminal ou na linha de comando, no local do diretório onde você deseja clonar o repositório, execute os comandos `git config --local user.name` e `git config --local user.email` para definir o nome do usuário e o e-mail para as confirmações que você fará no repositório. Por exemplo:

```
git config --local user.name "Saanvi Sarkar"
git config --local user.email saanvi_sarkar@example.com
```

Esses comandos não retornam nada, mas o e-mail e o nome de usuário especificados serão associados às confirmações feitas no repositório na AccountA.

4. Execute o `aws configure --profile` comando para configurar um perfil padrão a ser usado ao se conectar aos recursos na AccountB. Quando solicitado, forneça a chave de acesso e a chave secreta para seu usuário do IAM.

 Note

Se você já instalou AWS CLI e configurou um perfil, pode pular esta etapa.

Por exemplo, execute o comando a seguir para criar um AWS CLI perfil padrão que você usa para acessar AWS recursos no AccountB no Leste dos EUA (Ohio) (us-east-2):

```
aws configure
```

Quando solicitado, forneça as seguintes informações:

```
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 ID [None]: us-east-2  
Default output format [None]: json
```

5. Execute o comando `aws configure --profile` novamente para configurar um perfil nomeado a ser usado ao se conectar ao repositório na AccountA. Quando solicitado, forneça a chave de acesso e a chave secreta para seu usuário do IAM. Por exemplo, execute o comando a seguir para criar um AWS CLI perfil chamado *MyCrossAccountAccessProfile* que você usa para acessar um repositório na AccountA no Leste dos EUA (Ohio) (us-east-2):

```
aws configure --profile MyCrossAccountAccessProfile
```

Quando solicitado, forneça as seguintes informações:

```
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 ID [None]: us-east-2  
Default output format [None]: json
```

6. Em um editor de texto não criptografado, abra o arquivo `config`, também conhecido como o arquivo de configuração da AWS CLI. Dependendo do seu sistema operacional, esse arquivo pode estar localizado `~/.aws/config` no Linux, macOS ou Unix, ou em *drive*:\Users\ \.aws*USERNAME*\ `config` no Windows.
7. No arquivo, localize a entrada que corresponde ao perfil padrão configurado para acesso aos repositórios na AccountB. A aparência deve ser semelhante à seguinte:

```
[default]  
region = us-east-2  
output = json
```

Adicione `account` à configuração do perfil. Forneça o ID da conta da AWS da AccountB. Por exemplo:

```
[default]  
account = 888888888888  
region = us-east-2  
output = json
```

8. No arquivo, encontre a entrada que corresponde ao *MyCrossAccountAccessProfile* perfil que você acabou de criar. A aparência deve ser semelhante à seguinte:

```
[profile MyCrossAccountAccessProfile]  
region = us-east-2  
output = json
```

Adicione `account`, `role_arn` e `source_profile` à configuração do perfil. Forneça o ID da conta da Amazon Web Services da AccountA, o ARN da função na AccountA que você assume para acessar o repositório na outra conta e o nome do perfil padrão da AWS CLI na AccountB. Por exemplo:

```
[profile MyCrossAccountAccessProfile]  
region = us-east-2  
account = 111122223333  
role_arn = arn:aws:iam::111122223333:role/MyCrossAccountRepositoryContributorRole  
source_profile = default  
output = json
```

Salve suas alterações e feche o editor de texto plano.

Etapa 2: clonar e acessar o CodeCommit repositório na AccountA

Execute `git clone`, `git push` e `git pull` para clonar, enviar por push e efetuar pull do repositório do CodeCommit entre contas. Você também pode entrar no AWS Management Console, trocar de função e usar o CodeCommit console para interagir com o repositório na outra conta.

Note

Dependendo de como a função do IAM foi configurada, talvez você consiga visualizar os repositórios na página padrão do CodeCommit. Se você não conseguir visualizar os repositórios, peça ao administrador do repositório que envie por e-mail um link de URL para a página de código do repositório compartilhado no console. O URL é semelhante ao seguinte:

```
https://console.aws.amazon.com/codecommit/home?region=us-east-2#/repository/MySharedDemoRepo/browse/HEAD/--/
```


Para clonar o repositório entre várias contas para seu computador local

1. Na linha de comando ou no terminal, no diretório onde você deseja clonar o repositório, execute o comando git clone com o URL do clone (GRC) HTTPS. Por exemplo:

```
git clone codecommit://MyCrossAccountAccessProfile@MySharedDemoRepo
```

A não ser que você especifique o contrário, o repositório é clonado em um subdiretório com o mesmo nome do repositório.

2. Altere os diretórios do repositório clonado e adicione ou faça uma alteração em um arquivo. Por exemplo, você pode adicionar um arquivo chamado *NewFile.txt*.
3. Adicione o arquivo às alterações controladas do repositório local, confirme a alteração e envie o arquivo para o CodeCommit repositório. Por exemplo:

```
git add NewFile.txt
git commit -m "Added a file to test cross-account access to this repository"
git push
```

Para obter mais informações, consulte [Começando com o Git e AWS CodeCommit](#).

Agora que você adicionou um arquivo, acesse o CodeCommit console para ver seu commit, analisar as alterações de outros usuários no repositório, participar de pull requests e muito mais.

Para acessar o repositório de várias contas no console CodeCommit

1. Faça login Console de gerenciamento da AWS no AccountB (888888888888) como o usuário do IAM que recebeu acesso entre contas ao repositório no AccountA.
2. Escolha seu nome de usuário na barra de navegação e, na lista suspensa, escolha Switch Role.

 Note

Se esta for a primeira vez que você selecionou essa opção, revise as informações na página e, em seguida, escolha Switch Role novamente.

3. Na página Switch Role, faça o seguinte:
 - Em Account (Conta), insira o ID da conta para a AccountA (por exemplo, 111122223333).

- Em Role, insira o nome da função que você deseja assumir para acessar o repositório na AccountA (por exemplo *MyCrossAccountRepositoryContributorRole*,).
- Em Display Name (Nome de exibição), insira um nome amigável para essa função. Esse nome aparece no console ao assumir essa função. Ele também é exibido na lista de funções assumidas na próxima vez que você quiser alternar funções no console.
- (Opcional) Em Color, escolha um rótulo de cor para o nome de exibição.
- Selecione Mudar de perfil.

Para obter mais informações, consulte [Alternar para uma função \(Console de gerenciamento da AWS\)](#).

4. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.

Se a função assumida tiver permissão para visualizar os nomes dos repositórios na AccountA, você verá uma lista de repositórios e uma mensagem de erro que informa que você não tem permissões para visualizar o status deles. Esse comportamento é esperado. Escolha o nome do repositório compartilhado na lista.

Se a função assumida não tiver permissão para visualizar os nomes dos repositórios na AccountA, você verá uma mensagem de erro e uma lista em branco sem repositórios. Cole o link de URL do repositório ou modifique o link do console e altere `/list` para o nome do repositório compartilhado (por exemplo, */MySharedDemoRepo*).

5. Em Code, localize o nome do arquivo que você adicionou pelo computador local. Escolha-o para procurar pelo código no arquivo e, em seguida, procure no restante do repositório e comece a usar seus recursos.

Para obter mais informações, consulte [Começando com AWS CodeCommit](#).

Excluir um AWS CodeCommit repositório

Você pode usar o CodeCommit console ou o AWS CLI para excluir um CodeCommit repositório.

 Note

A exclusão de um repositório não exclui nenhuma cópia local do repositório (repositórios locais). Para excluir um repositório local, use o diretório da sua máquina local e as ferramentas de gerenciamento de arquivo.

Tópicos

- [Excluir um CodeCommit repositório \(console\)](#)
- [Excluir um repositório local](#)
- [Excluir um CodeCommit repositório \(AWS CLI\)](#)

Excluir um CodeCommit repositório (console)

Siga estas etapas para usar o CodeCommit console para excluir um CodeCommit repositório.

 Important

Depois de excluir um CodeCommit repositório, você não poderá mais cloná-lo em nenhum repositório local ou compartilhado. Também não será mais possível extrair dados dele nem enviar via push dados para ele de nenhum repositório local ou compartilhado. Esta ação não pode ser desfeita.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome daquele que você deseja excluir.
3. No painel de navegação, selecione Configurações.
4. Na guia General (Geral), em Delete repository (Excluir repositório), selecione Delete repository (Excluir repositório). Insira **delete** e escolha Delete (Excluir). O repositório será excluído permanentemente.

 Note

A exclusão do repositório em CodeCommit não exclui nenhum repositório local.

Excluir um repositório local

Use o diretório da sua máquina local e as ferramentas de gerenciamento de arquivo para excluir o diretório que contém o repositório local.

A exclusão de um repositório local não exclui nenhum CodeCommit repositório ao qual ele possa estar conectado.

Excluir um CodeCommit repositório (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para excluir um CodeCommit repositório, execute o `delete-repository` comando, especificando o nome do CodeCommit repositório a ser excluído (com a `--repository-name` opção).

Important

Depois de excluir um CodeCommit repositório, você não poderá mais cloná-lo em nenhum repositório local ou compartilhado. Também não será mais possível extrair dados dele nem enviar via push dados para ele de nenhum repositório local ou compartilhado. Esta ação não pode ser desfeita.

Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

Por exemplo, para excluir um repositório chamado MyDemoRepo:

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

Se for bem-sucedido, o ID do CodeCommit repositório que foi excluído permanentemente aparecerá na saída:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
```

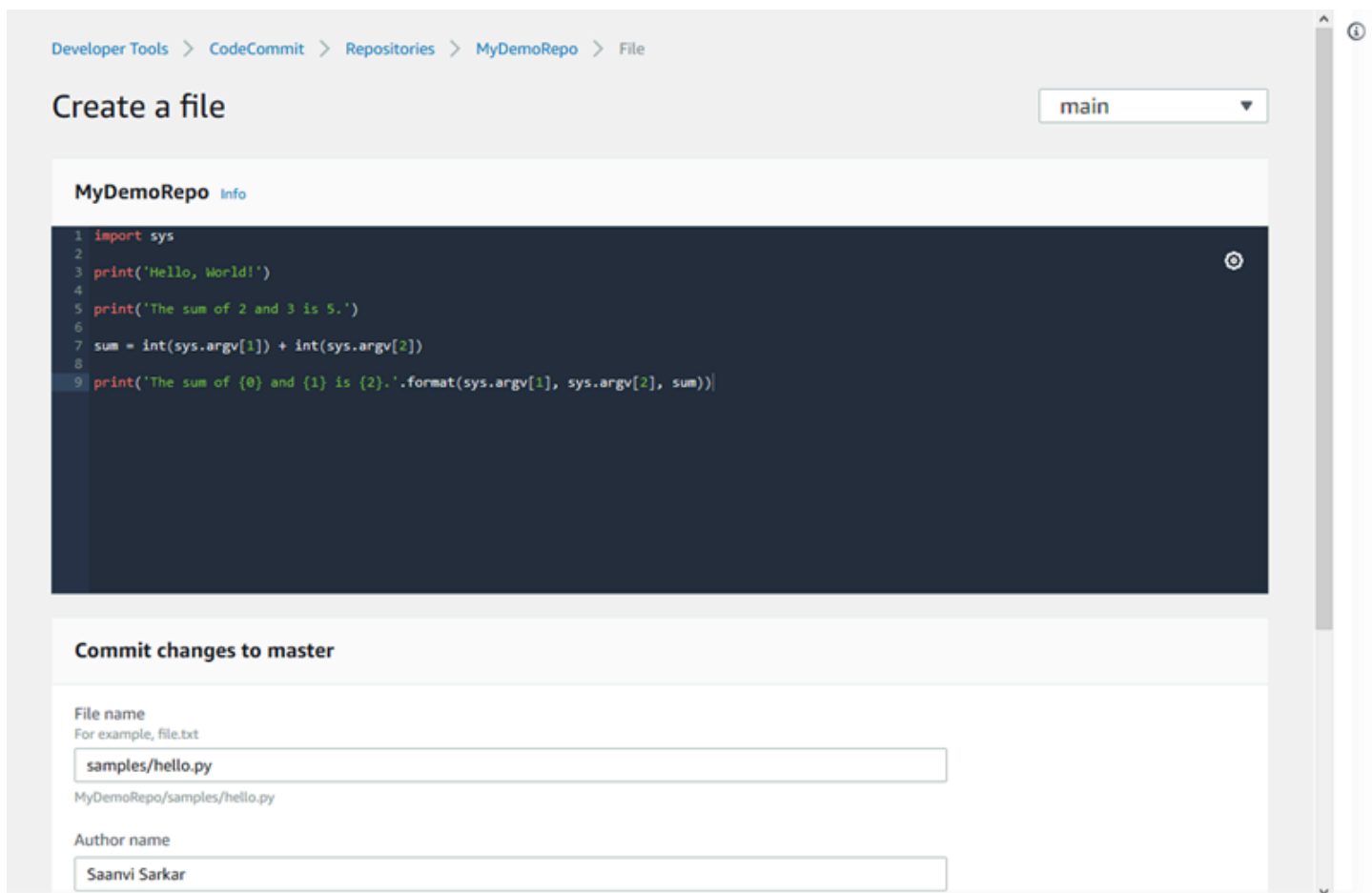
```
}
```

A exclusão de um CodeCommit repositório não exclui nenhum repositório local que possa estar conectado a ele.

Trabalhar com arquivos em repositórios do AWS CodeCommit

No CodeCommit, um arquivo é uma informação autônoma e controlada pela versão disponível para você e outros usuários do repositório e ramificação nos quais o arquivo está armazenado. Você pode organizar seus arquivos de repositório em uma estrutura de diretórios, assim como faria em um computador. Ao contrário de um computador, o CodeCommit rastreia automaticamente todas as alterações em um arquivo. Você pode comparar as versões de um arquivo e armazenar diferentes versões dele em diferentes ramificações do repositório.

Para adicionar ou editar um arquivo em um repositório, você pode usar um cliente do Git. Você também pode usar o console do CodeCommit, a AWS CLI, ou a API do CodeCommit.



Para obter informações sobre como trabalhar com outros aspectos do seu repositório no CodeCommit, consulte [Trabalhar com repositórios](#), [Trabalhar com solicitações pull](#), [Como trabalhar com ramificações](#), [Trabalhar com confirmações](#) e [Trabalhar com preferências do usuário](#).

Tópicos

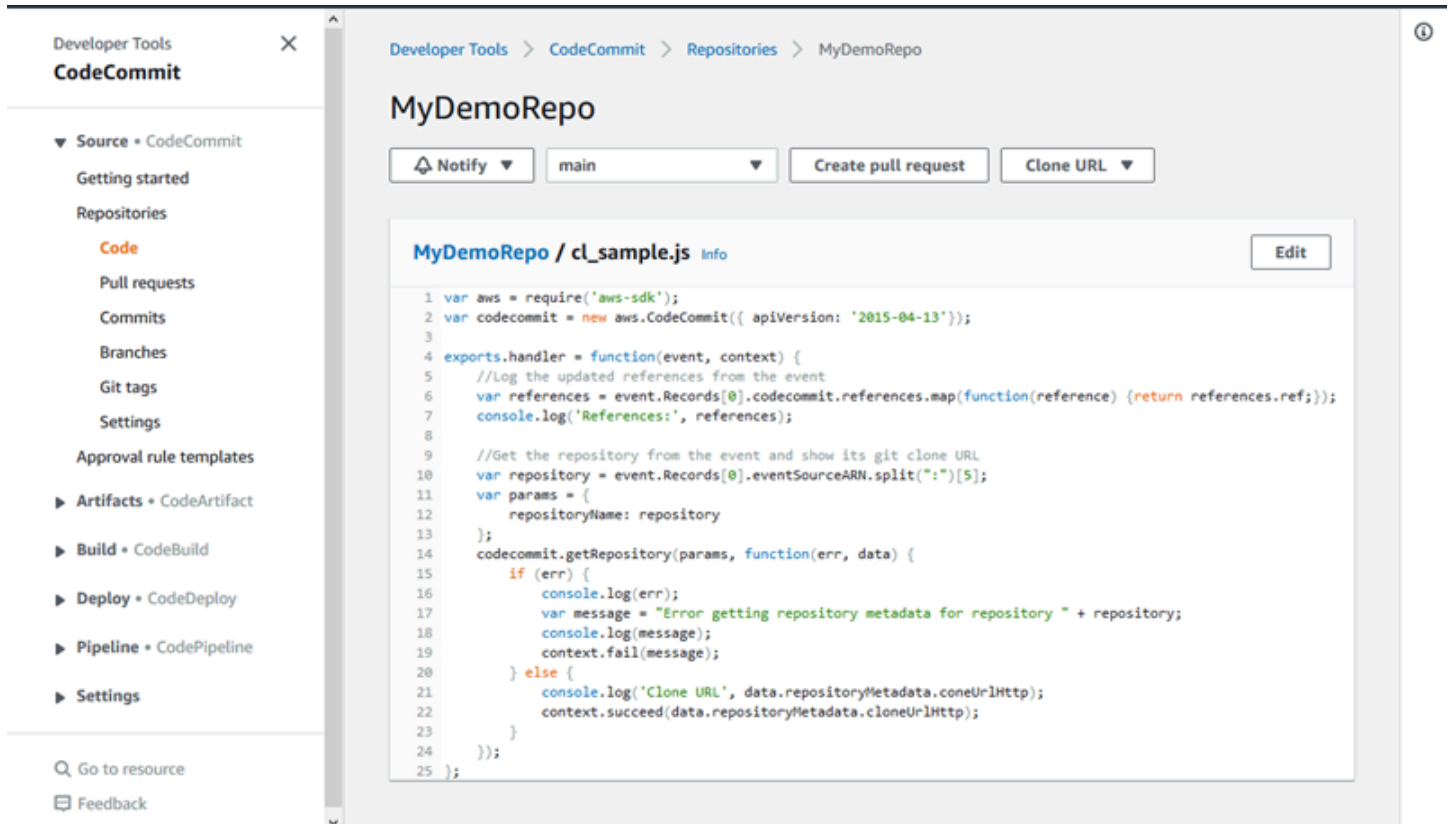
- [Procurar arquivos em um AWS CodeCommit repositório](#)
- [Criar ou adicionar um arquivo para um repositório do AWS CodeCommit](#)
- [Editar o conteúdo de um arquivo em um AWS CodeCommit repositório](#)

Procurar arquivos em um AWS CodeCommit repositório

Depois de se conectar a um CodeCommit repositório, você pode cloná-lo em um repositório local ou usar o CodeCommit console para pesquisar seu conteúdo. Este tópico descreve como usar o CodeCommit console para pesquisar o conteúdo de um CodeCommit repositório.

Note

Para CodeCommit usuários ativos, não há cobrança pelo código de navegação no CodeCommit console. Para obter informações sobre quando podem ser aplicadas cobranças, consulte [Definição de preço](#).



Navegue em um CodeCommit repositório

Você pode usar o CodeCommit console para revisar os arquivos contidos em um repositório ou para ler rapidamente o conteúdo de um arquivo.

Para navegar no conteúdo de um repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Na página Repositories (Repositórios), na lista de repositórios, escolha o repositório pelo qual deseja navegar.
3. Na visualização Code, navegue pelo conteúdo da ramificação padrão de seu repositório.

Para alterar a visualização para outra ramificação ou tag, escolha o botão seletor de visualização. Escolha um nome de ramificação ou de tag na lista suspensa ou, na caixa de filtro, insira o nome da ramificação ou da tag e escolha-o na lista.

4. Execute um destes procedimentos:
 - Para visualizar o conteúdo de um diretório, escolha-o na lista. Você pode escolher alguns diretórios na lista de navegação para retornar à visualização do diretório. Também é possível usar a seta para cima na parte superior da lista de diretórios.
 - Como visualizar o conteúdo de um arquivo, escolha-o na lista. Se o arquivo for maior do que o limite de objeto de confirmação, ele não poderá ser visualizado no console e deve ser visualizado em um repositório local. Para obter mais informações, consulte [Cotas](#). Para sair da visualização do arquivo, na barra de navegação de código, escolha o diretório que deseja ver.

Note

Nem todos os arquivos binários podem ser visualizados no console. Se você escolher um arquivo binário potencialmente visualizável, será exibida uma mensagem de aviso pedindo que você confirme se deseja exibir o conteúdo. Para visualizar o arquivo, escolha Mostrar conteúdo do arquivo. Caso não queira ver o arquivo, na barra de navegação de código, escolha o diretório que deseja visualizar.

Se você escolher um arquivo Markdown (.md), use os botões Markdown renderizado e Origem do Markdown para alternar entre as visualizações renderizada e de sintaxe. Para obter mais informações, consulte [Usar Markdown no console](#).

Criar ou adicionar um arquivo para um repositório do AWS CodeCommit

Você pode usar o console do CodeCommit, a AWS CLI ou um cliente Git para adicionar um arquivo a um repositório. Você pode fazer upload de um arquivo do computador local no repositório, ou usar o editor de códigos no console para criar o arquivo. O editor é uma maneira rápida e fácil de adicionar um arquivo simples, como um arquivo `readme.md`, a uma ramificação de um repositório.

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**

Tópicos

- [Criar ou fazer upload de um arquivo \(console\)](#)
- [Adicionar um arquivo \(AWS CLI\)](#)
- [Adicionar um arquivo \(Git\)](#)

Criar ou fazer upload de um arquivo (console)

Você pode usar o console do CodeCommit para criar ou adicionar um arquivo a uma ramificação em um repositório do CodeCommit. Como parte da criação do arquivo, você pode fornecer seu nome de usuário e um endereço de e-mail. Também é possível adicionar uma mensagem de confirmação para os outros usuários saberem que você adicionou o arquivo e por qual motivo. Você também

pode fazer upload de um arquivo diretamente do computador local para uma ramificação em um repositório.

Para adicionar um arquivo a um repositório

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório que onde deseja adicionar um arquivo.
3. Na visualização Code, escolha a ramificação na qual deseja adicionar o arquivo. Por padrão, o conteúdo da ramificação padrão é mostrado quando você abre a visualização Code.

Para alterar a visualização para outra ramificação, escolha o botão seletor de visualização. Escolha um nome de ramificação na lista suspensa ou, na caixa de filtro, insira o nome da ramificação e escolha-o na lista.

4. Escolha Add file e, em seguida, escolha uma das seguintes opções:
 - Para usar o editor de códigos para criar o conteúdo de um arquivo e adicioná-lo ao repositório, selecione Create file (Criar arquivo).
 - Para fazer upload de um arquivo do computador local para o repositório, selecione Upload file (Fazer upload de arquivo).
5. Forneça informações a outros usuários sobre quem adicionou esse arquivo ao repositório e por qual motivo.
 - Em Author name (Nome do autor), insira seu nome. Esse nome é usado como o nome do autor e do responsável pela confirmação nas informações de confirmação. Por padrão, o CodeCommit usa seu nome de usuário do IAM ou uma derivação do seu login no console como nome do autor.
 - Em Endereço de e-mail, insira um endereço de e-mail para que outros usuários do repositório possam entrar em contato sobre essa alteração.
 - Em Commit message (Confirmar mensagem), insira uma breve descrição. Isso é opcional, mas altamente recomendado. Caso contrário, uma mensagem de confirmação padrão será usada.
6. Execute um destes procedimentos:
 - Se você estiver fazendo upload de um arquivo, escolha o arquivo no seu computador local.
 - Se você estiver criando um arquivo, insira o conteúdo que deseja adicionar no editor de códigos e dê um nome para o arquivo.

7. Escolha Commit changes (Confirmar alterações).

Adicionar um arquivo (AWS CLI)

Você pode usar a AWS CLI e o comando `put-file` para adicionar um arquivo em um repositório do CodeCommit. Você também pode usar o comando `put-file` para adicionar um diretório ou uma estrutura de caminho ao arquivo.

Note

Para usar comandos da AWS CLI com o CodeCommit, instale a AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Para adicionar um arquivo a um repositório

1. No computador local, crie o arquivo que você deseja adicionar ao repositório do CodeCommit.
2. No terminal ou na linha de comando, execute o comando `put-file`, especificando:
 - O repositório no qual você deseja adicionar o arquivo.
 - A ramificação na qual você deseja adicionar o arquivo.
 - O ID de confirmação completo da confirmação mais recente feita nessa ramificação, também conhecido como a confirmação de ponta ou cabeça.
 - A localização atual do arquivo. A sintaxe usada para essa localização varia, dependendo do seu sistema operacional local.
 - O nome do arquivo que você deseja adicionar, incluindo o caminho onde o arquivo atualizado está armazenado no repositório, se houver.
 - O nome de usuário e e-mail a serem associados a este arquivo.
 - Uma mensagem de confirmação explicando por que você adicionou este arquivo.

O nome de usuário, endereço de e-mail e a mensagem de confirmação são opcionais, mas ajudam outros usuários a saber quem fez a alteração e por quê. Se você não fornecer um nome de usuário, o CodeCommit usará seu nome de usuário do IAM como padrão ou uma derivação do seu login no console como nome do autor.

Por exemplo, para adicionar um arquivo chamado *SoluçãoDeExemplo.py* a um repositório chamado *MeuRepoDemo* para uma ramificação denominada *recurso-recursodealeatorização* cuja confirmação mais recente apresenta um ID de *4c925148EXEMPLO*:

```
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 você adiciona arquivos binários, use `fileb://` para especificar a localização atual do arquivo.

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

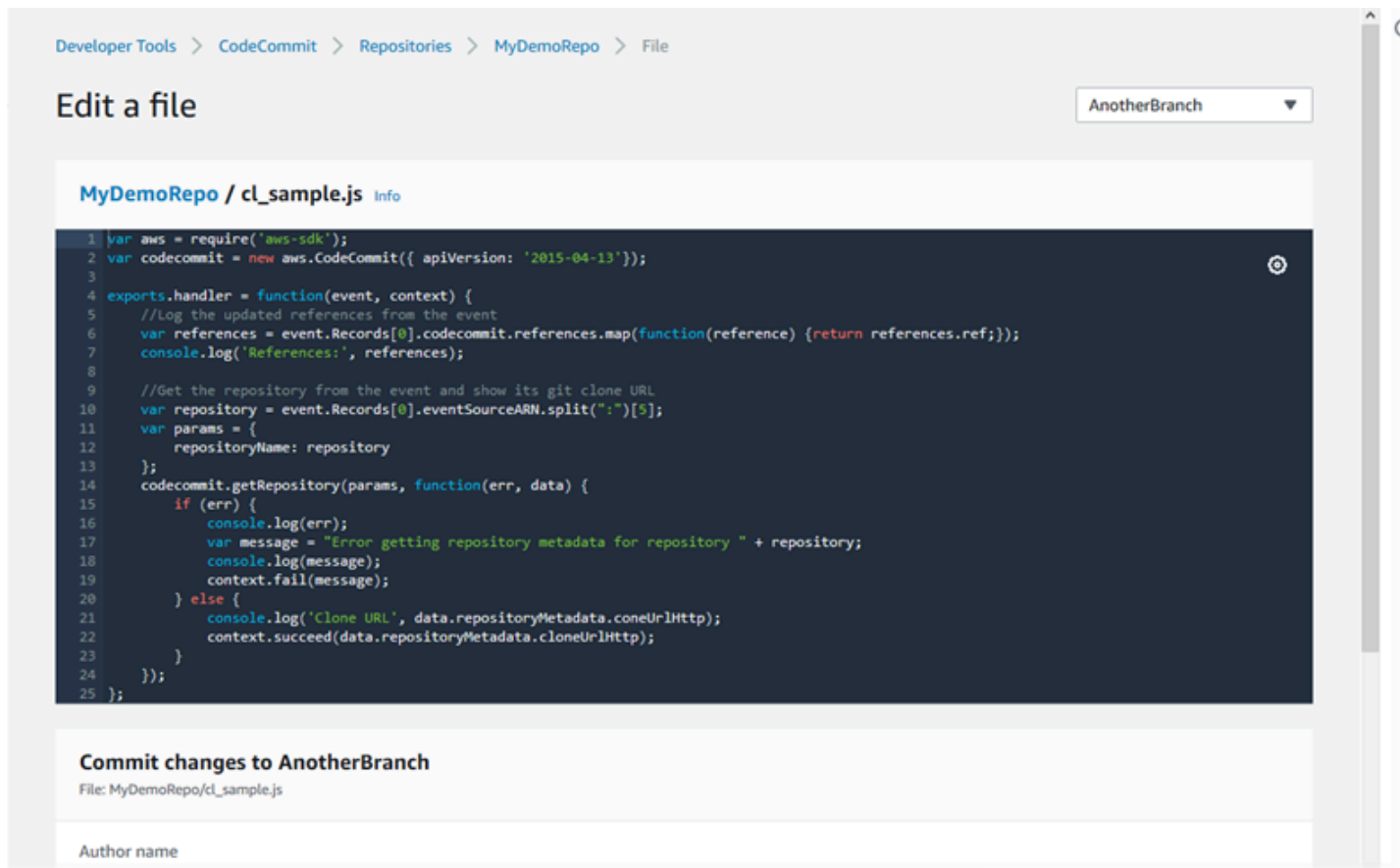
```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Adicionar um arquivo (Git)

Você pode adicionar arquivos a um repositório local e enviar as alterações via push a um repositório do CodeCommit. Para obter mais informações, consulte [Começando com o Git e AWS CodeCommit](#).

Editar o conteúdo de um arquivo em um AWS CodeCommit repositório

Você pode usar o CodeCommit console ou um cliente Git para editar o conteúdo de um arquivo em um CodeCommit repositório. AWS CLI



Tópicos

- [Editar um arquivo \(console\)](#)
- [Editar ou excluir um arquivo \(AWS CLI\)](#)
- [Editar um arquivo \(Git\)](#)

Editar um arquivo (console)

Você pode usar o CodeCommit console para editar um arquivo que foi adicionado a uma ramificação em um CodeCommit repositório. Como parte da edição do arquivo, você pode fornecer seu nome de usuário e um endereço de e-mail. Também é possível adicionar uma mensagem de confirmação para os outros usuários saberem que você fez a alteração e por quê.

Para editar um arquivo em um repositório

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório que onde deseja editar um arquivo.

3. Na visualização Code, escolha a ramificação na qual deseja editar o arquivo. Por padrão, o conteúdo da ramificação padrão é mostrado quando você abre a visualização Code.

Para alterar a visualização para outra ramificação, escolha o botão seletor de visualização. Escolha um nome de ramificação na lista suspensa ou, na caixa de filtro, insira o nome da ramificação e escolha-o na lista.

4. Navegue pelo conteúdo da ramificação e escolha o arquivo que deseja editar. Na visualização, escolha Edit.

 Note

Se você escolher um arquivo binário, será exibida uma mensagem de aviso pedindo que você confirme se deseja exibir o conteúdo. Você não deve usar o CodeCommit console para editar arquivos binários.

5. Edite o arquivo e forneça informações a outros usuários sobre quem fez essa alteração e por qual motivo.
 - Em Author name (Nome do autor), insira seu nome. Esse nome é usado como nome do autor e nome do confirmador nas informações do commit. CodeCommit usa como padrão o nome de usuário do IAM ou uma derivação do login do console como nome do autor.
 - Em Endereço de e-mail, insira um endereço de e-mail para que outros usuários do repositório possam entrar em contato sobre essa alteração.
 - Em Commit message (Mensagem de confirmação), insira uma breve descrição de suas alterações.
6. Escolha Commit changes para salvar as alterações no arquivo e confirmar as alterações no repositório.

Editar ou excluir um arquivo (AWS CLI)

Você pode usar o put-file comando AWS CLI and the para fazer alterações em um arquivo em um CodeCommit repositório. Você também pode usar o comando put-file para adicionar um diretório ou estrutura de caminho ao arquivo alterado, caso deseje armazenar o arquivo alterado em um local diferente do original. Se quiser excluir um arquivo completamente, você poderá usar o comando delete-file.

 Note

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para editar um arquivo em um repositório

1. Usando uma cópia local do arquivo, faça as alterações que deseja adicionar ao repositório do CodeCommit.
2. No terminal ou na linha de comando, execute o comando `put-file`, especificando:
 - O repositório no qual você deseja adicionar o arquivo editado.
 - A ramificação na qual você deseja adicionar o arquivo editado.
 - O ID de confirmação completo da confirmação mais recente feita nessa ramificação, também conhecido como a confirmação de ponta ou cabeça.
 - A localização atual do arquivo.
 - O nome do arquivo atualizado que você deseja adicionar, incluindo o caminho onde o arquivo atualizado está armazenado no repositório, se houver.
 - O nome de usuário e e-mail a serem associados a esta alteração.
 - Uma mensagem de confirmação que explica a alteração que você fez.

O nome de usuário, endereço de e-mail e a mensagem de confirmação são opcionais, mas ajudam outros usuários a saber quem fez a alteração e por quê. Se você não fornecer um nome de usuário, o CodeCommit padrão é usar seu nome de usuário do IAM ou uma derivação do seu login do console.

Por exemplo, para adicionar edições feitas em um arquivo chamado *ExampleSolution.py* a um repositório chamado *MyDemoRepo* a uma ramificação chamada *feature-randomizationfeature* cujo commit mais recente tem um ID de: *4c925148EXAMPLE*

```
aws codecommit put-file --repository-name MyDemoRepo --branch-name feature-  
randomizationfeature --file-content file://MyDirectory/ExampleSolution.py --file-  
path /solutions/ExampleSolution.py --parent-commit-id 4c925148EXAMPLE --name "María  
García" --email "maría_garcía@example.com" --commit-message "I fixed the bug Mary  
found."
```

 Note

Se você deseja adicionar um arquivo binário alterado, use `--file-content` com a notação **fileb://***MyDirectory/MyFile.raw*.

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

Para excluir um arquivo, use o comando `delete-file`. Por exemplo, para excluir um arquivo nomeado *README.md* em uma ramificação chamada *main* com uma ID de confirmação mais recente ou *c5709475EXAMPLE* em um repositório chamado *MyDemoRepo*:

```
aws codecommit delete-file --repository-name MyDemoRepo --branch-name main --file-
path README.md --parent-commit-id c5709475EXAMPLE
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "blobId": "559b44fEXAMPLE",
  "commitId": "353cf655EXAMPLE",
  "filePath": "README.md",
  "treeId": "6bc824cEXAMPLE"
}
```

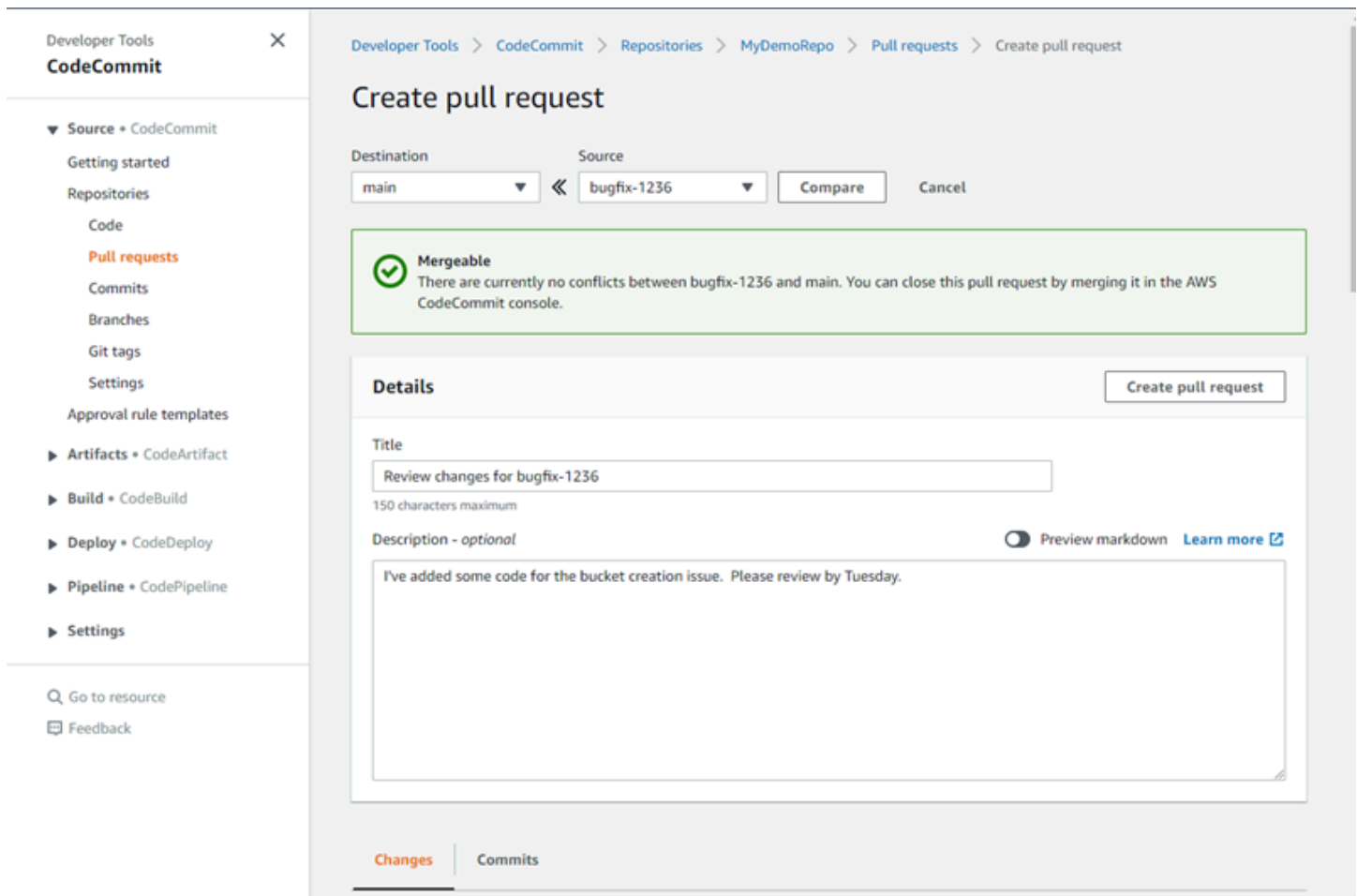
Editar um arquivo (Git)

Você pode editar arquivos em um repositório local e enviar suas alterações para um CodeCommit repositório. Para obter mais informações, consulte [Começando com o Git e AWS CodeCommit](#).

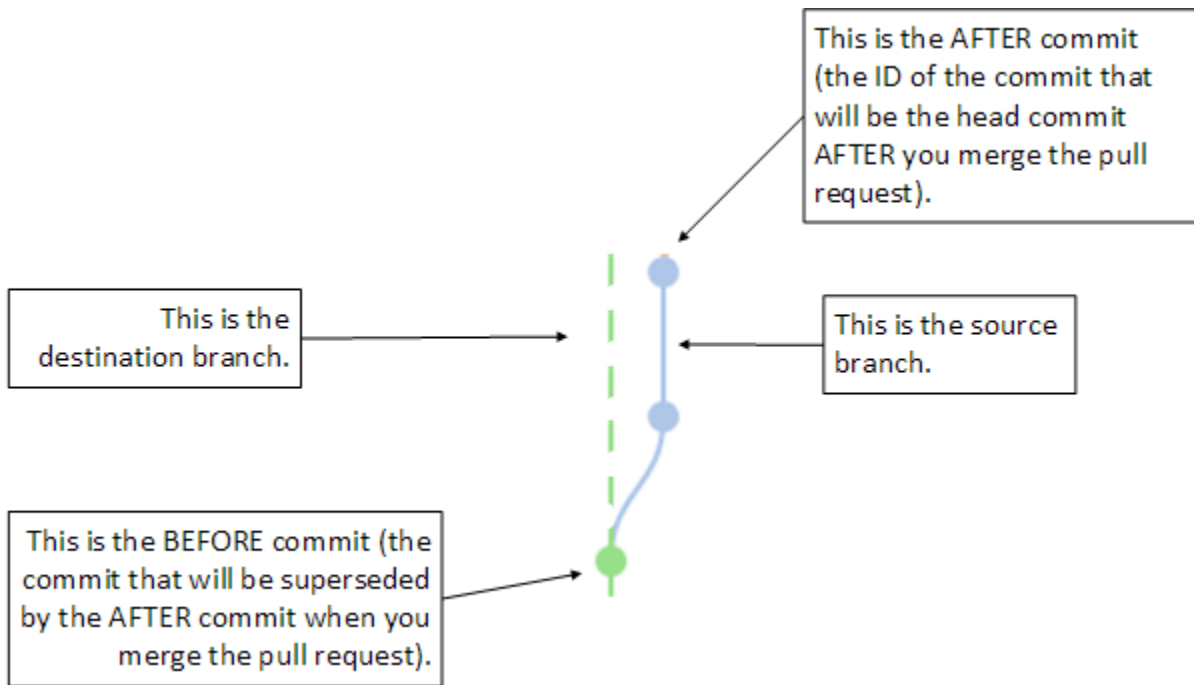
Trabalhando com pull requests em AWS CodeCommit repositórios

Uma solicitação de extração é a principal maneira para você e outros usuários de repositório analisarem, comentarem e mesclarem alterações de código de uma ramificação para outra. Você pode usar solicitações de extração para analisar alterações de código de maneira colaborativa em caso de alterações ou correções relativamente pequenas, adições significativas de recursos ou novas versões do seu software lançado. Aqui está um possível fluxo de trabalho para uma solicitação pull:

Li Juan, um desenvolvedor que trabalha em um repositório chamado MyDemoRepo, quer trabalhar em um novo recurso para uma próxima versão de um produto. Para manter seu trabalho separado do código pronto para produção, ela cria uma ramificação a partir da ramificação padrão e a nomeia. *feature-randomizationfeature* Ela escreve código, faz confirmações e envia o novo código de recurso por push para essa ramificação. Ela quer que outros usuários do repositório analisem o código para verificar a qualidade antes de ela mesclar suas alterações na ramificação padrão. Para isso, ela cria uma solicitação pull. A solicitação de extração contém a comparação entre sua ramificação de trabalho e a ramificação do código em que ela pretende mesclar as alterações (neste caso, a ramificação padrão). Ela também pode criar uma regra de aprovação que exige que um número especificado de usuários aprovem a solicitação pull. Ela pode até mesmo especificar um grupo de aprovações de usuários. Outros usuários analisam seu código e alterações adicionando comentários e sugestões. Ela pode atualizar sua ramificação de trabalho várias vezes com alterações de código em resposta aos comentários. Suas alterações são incorporadas ao pull request toda vez que ela as envia para essa ramificação. CodeCommit Ela também pode incorporar as alterações que foram feitas na ramificação de destino pretendida enquanto a solicitação de extração estiver aberta, para que os usuários possam ter certeza de que estão revisando todas as alterações propostas no contexto. Quando ela e seus revisores estiverem satisfeitos e as condições para as regras de aprovação (se houver) forem atendidas, ela ou um de seus revisores mescla o código e fecha a solicitação pull.



As solicitações de extração exigem duas ramificações: uma ramificação de origem, que contém o código que você deseja analisar, e uma ramificação de destino, na qual você mescla o código revisado. A ramificação de origem contém a confirmação AFTER, que é a confirmação que contém as alterações que você deseja mesclar na ramificação de destino. A ramificação de destino contém a confirmação BEFORE, que representa o estado do código (antes de a ramificação da solicitação pull ser mesclada à ramificação de destino). A escolha da estratégia de mesclagem afeta os detalhes de como os commits são mesclados entre as ramificações de origem e destino no console. Para obter mais informações sobre estratégias de mesclagem em CodeCommit, consulte [Mesclar uma solicitação pull \(console\)](#).



A solicitação de extração exibe as diferenças entre a extremidade da ramificação de origem e a confirmação mais recente na ramificação de destino quando a solicitação de extração é criada para que os usuários possam visualizar e comentar sobre as alterações. Você pode atualizar a solicitação de extração em resposta a comentários confirmando e enviando alterações para a ramificação de origem.

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 >

☒ Hide whitespace changes ☐ Unified ☒ Split

ahs_count.py

```

*** @ -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`.

Quando o código for revisado e os requisitos da regra de aprovação (se houver) forem atendidos, será possível fechar a solicitação pull de várias maneiras:

- Mescle as ramificações localmente e faça push de suas alterações. Isso fecha a solicitação automaticamente se a estratégia de mesclagem encaminhada for usada e não houver conflitos de mesclagem.
- Use o AWS CodeCommit console para fechar a pull request sem mesclar, resolver conflitos em uma mesclagem ou, se não houver conflitos, fechar e mesclar as ramificações usando uma das estratégias de mesclagem disponíveis.
- Use a AWS CLI.

Antes de criar uma solicitação pull:

- Verifique se você confirmou e enviou as alterações de código por push que você deseja que sejam revisadas a uma ramificação (a ramificação de origem).
- Configure notificações para seu repositório para que outros usuários possam ser notificados sobre a solicitação de extração e alterações feitas a ela. (Esta etapa é opcional, mas recomendada.)
- Crie e associe modelos de regra de aprovação ao repositório para que as regras de aprovação sejam criadas automaticamente para solicitações pull a fim de ajudar a garantir a qualidade do código. Para obter mais informações, consulte [Trabalhar com modelos de regra de aprovação](#).

As solicitações pull são mais eficientes quando você configura usuários do IAM para os usuários do repositório na conta da Amazon Web Services. É mais fácil identificar qual usuário fez qual comentário. A outra vantagem é que os usuários do IAM podem usar credenciais do Git para acessar o repositório. Para obter mais informações, consulte [Etapa 1: Configuração inicial para CodeCommit](#). Você pode usar solicitações de extração com outros tipos de usuários, incluindo usuários de acesso federado.

Para obter informações sobre como trabalhar com outros aspectos do seu repositório em CodeCommit [Trabalhar com repositórios](#), consulte [Trabalhar com modelos de regra de aprovação](#), [Trabalhar com arquivos](#), [Trabalhar com confirmações](#), [Como trabalhar com ramificações](#), e [Trabalhar com preferências do usuário](#)

Tópicos

- [Criar uma solicitação pull](#)
- [Criar uma regra de aprovação de uma solicitação pull](#)
- [Exibir pull requests em um AWS CodeCommit repositório](#)
- [Revisar uma solicitação pull](#)

- [Atualizar uma solicitação pull](#)
- [Editar ou excluir uma regra de aprovação de uma solicitação pull](#)
- [Substituir regras de aprovação em uma solicitação pull](#)
- [Mesclar uma solicitação pull em um repositório do AWS CodeCommit](#)
- [Resolver conflitos em uma solicitação pull em um repositório do AWS CodeCommit](#)
- [Fechar uma pull request em um AWS CodeCommit repositório](#)

Criar uma solicitação pull

A criação de solicitações pull ajuda outros usuários a ver e revisar suas alterações de código antes de mesclá-las em outra ramificação. Primeiro, você cria uma ramificação para suas alterações de código. Ela é chamada de ramificação de origem para uma solicitação pull. Depois de confirmar e enviar alterações para o repositório, você pode criar uma solicitação pull que compara o conteúdo dessa ramificação (a ramificação de origem) com a ramificação em que deseja mesclar suas alterações após a solicitação pull ser fechada (a ramificação de destino).

Você pode usar o console do AWS CodeCommit ou o AWS CLI para criar solicitações de extração para o repositório.

Tópicos

- [Criar uma solicitação pull \(console\)](#)
- [Criar uma solicitação pull \(AWS CLI\)](#)

Criar uma solicitação pull (console)

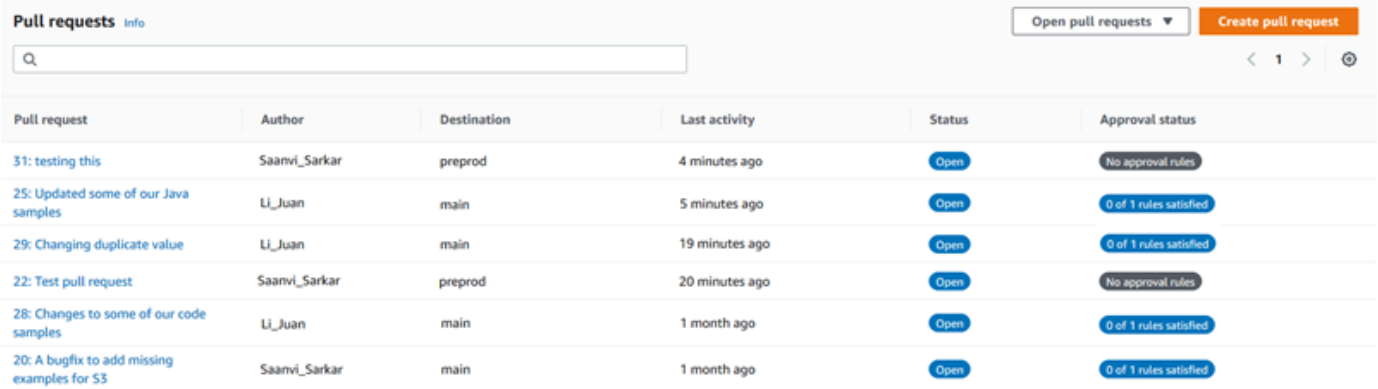
Você pode usar o console do CodeCommit para criar uma solicitação pull em um repositório do CodeCommit. Se o seu repositório estiver [configurado com notificações](#), os usuários inscritos receberão um e-mail quando você criar uma solicitação pull.

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja criar uma solicitação pull.
3. No painel de navegação, escolha Pull Requests.

Tip

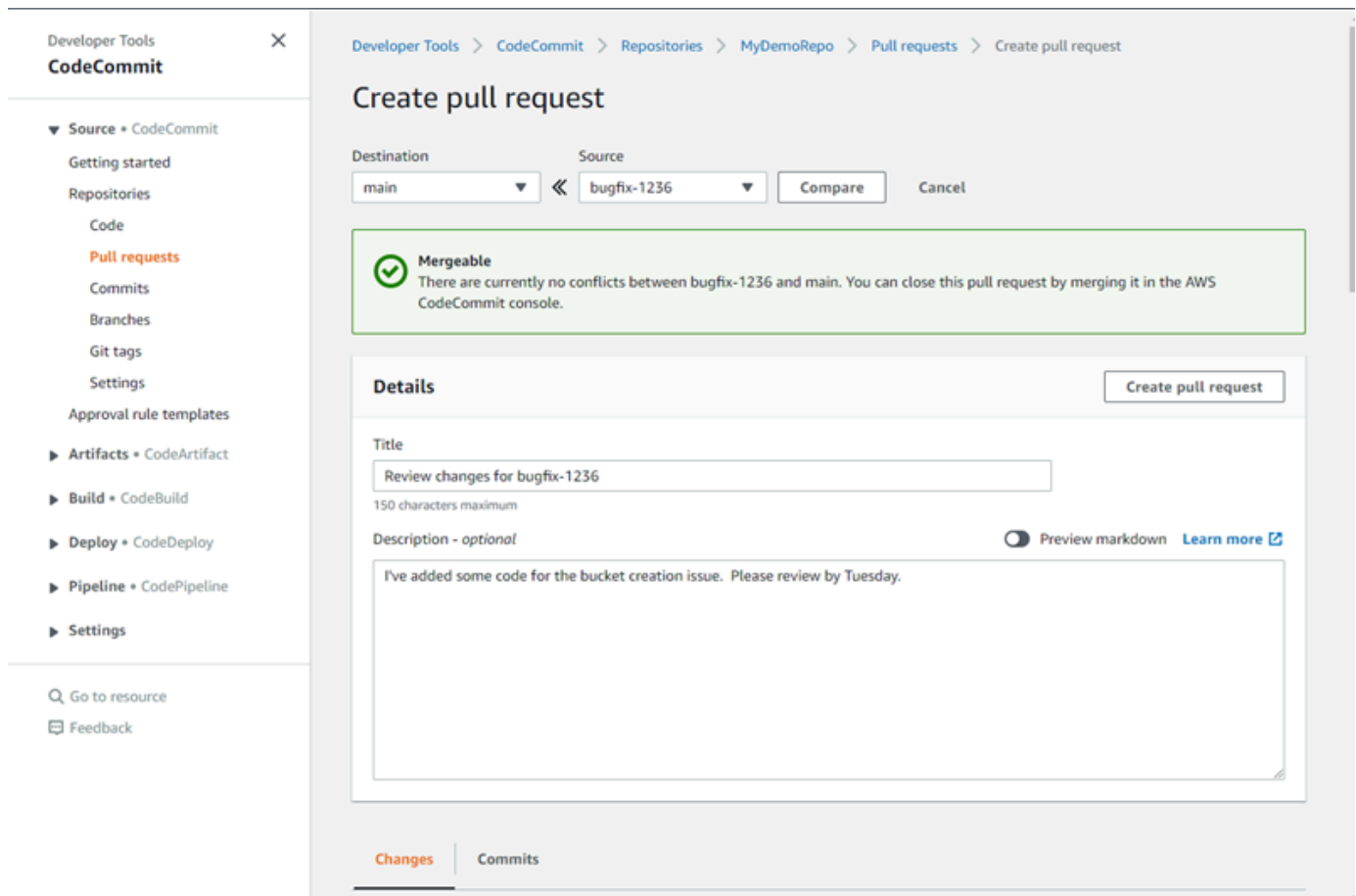
Você também pode criar solicitações pull em Branches (Ramificações) e em Code (Código).

4. Escolha 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

- Em Create pull request, em Source, escolha a ramificação que contém as alterações que deseja revisar.
- Em Destination (Destino), escolha a ramificação na qual você pretende mesclar suas alterações de código quando a solicitação pull for fechada.
- Escolha Compare. Uma comparação é executada nas duas ramificações, e as diferenças entre elas são exibidas. Uma análise também é realizada para determinar se as duas ramificações podem ser mescladas automaticamente quando a solicitação pull é fechada.
- Analise os detalhes da comparação e as alterações para se certificar de que a solicitação pull contém as alterações e confirmações que você deseja analisar. Caso contrário, ajuste suas escolhas para as ramificações de origem e de destino e escolha Compare novamente.
- Quando estiver satisfeito com os resultados da comparação para a solicitação pull, em Title (Título), forneça um título curto, mas descritivo, para essa revisão. Este é o título que aparece na lista de solicitações pull para o repositório.
- (Opcional) Em Description (Descrição), insira os detalhes sobre essa análise e qualquer outra informação útil para os revisores.
- Escolha Criar.



Sua solicitação pull aparece na lista de solicitações pull para o repositório. Se você [configurou notificações](#), os assinantes do tópico do Amazon SNS receberão um e-mail informando sobre a solicitação pull recém-criada.

Criar uma solicitação pull (AWS CLI)

Para usar comandos da AWS CLI com o CodeCommit, instale a AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar a AWS CLI para criar solicitações pull em um repositório do CodeCommit

- Execute o comando `create-pull-request`, especificando:
 - O nome da solicitação pull (com a opção `--title`).
 - A descrição da solicitação pull (com a opção `--description`).
 - Uma lista de destinos para o comando `create-pull-request`, incluindo:

- O nome do repositório do CodeCommit em que a solicitação pull foi criada (com o atributo `repositoryName`).
- O nome da ramificação que contém as alterações de código que você deseja analisar, também conhecida como ramificação de origem (com o atributo `sourceReference`).
- Opcionalmente, o nome da ramificação na qual você pretende mesclar suas alterações de código, também conhecida como a ramificação de destino, se você não quiser mesclar a ramificação padrão (com o atributo `destinationReference`).
- Um token de idempotência exclusivo gerado pelo cliente (com a opção `--client-request-token`).

Este exemplo cria uma solicitação pull chamada *Pronunciation difficulty analyzer* (*Analisador de dificuldade de pronúncia*) com uma descrição *Please review these changes by Tuesday* (*Revise essas alterações até terça-feira*) que tem como destino a ramificação de origem *jane-branch*. A solicitação pull deve ser mesclada com a ramificação padrão *principal* em um repositório do CodeCommit chamado *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
```

Criar uma regra de aprovação de uma solicitação pull

A criação de regras de aprovação de solicitações pull ajuda a garantir a qualidade do seu código, exigindo que os usuários aprove a solicitação pull para que o código possa ser mesclado à ramificação de destino. É possível especificar o número de usuários que devem aprovar uma solicitação pull. Também é possível especificar um grupo de aprovações de usuários para a regra. Se você fizer isso, somente as aprovações desses usuários contarão para o número de aprovações necessárias para a regra.

Note

Também é possível criar modelos de regras de aprovação que podem ajudar a automatizar a criação de regras de aprovação em vários repositórios. Para obter mais informações, consulte [Trabalhar com modelos de regra de aprovação](#).

Você pode usar o AWS CodeCommit console ou o AWS CLI para criar regras de aprovação para seu repositório.

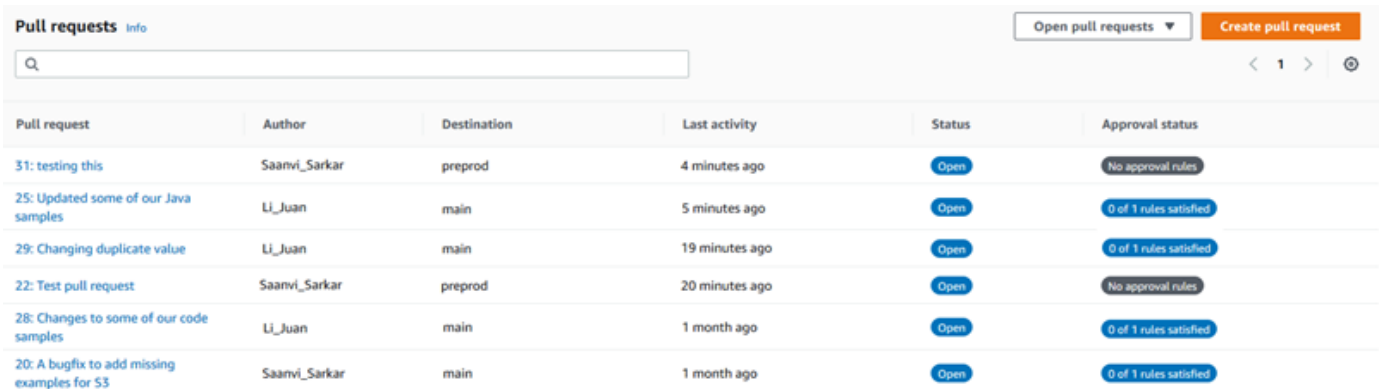
Tópicos

- [Criar uma regra de aprovação de uma solicitação pull \(console\)](#)
- [Criar uma regra de aprovação de uma solicitação pull \(AWS CLI\)](#)

Criar uma regra de aprovação de uma solicitação pull (console)

Você pode usar o CodeCommit console para criar uma regra de aprovação para uma pull request em um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja criar uma regra de aprovação de uma solicitação pull.
3. No painel de navegação, escolha Pull Requests.
4. Escolha a solicitação pull da qual você deseja criar uma regra de aprovação na lista. Só é possível criar regras de aprovação de solicitações pull abertas.



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. Na solicitação pull, escolha Approvals (Aprovações) e selecione Create approval rule (Criar regra de aprovação).
6. Em Rule name (Nome da regra), dê à regra um nome descritivo para que você saiba para que serve. Por exemplo, se você deseja exigir que duas pessoas aprovem uma solicitação pull para que ela possa ser mesclada, é possível nomear a regra **Require two approvals before merge**.

Note

Não é possível alterar o nome de uma regra de aprovação depois de criá-la.

Em Number of approvals needed (Número de aprovações necessárias), insira o número desejado. O padrão é um.

The screenshot shows the 'Create approval rule' interface in the AWS CodeCommit console. It features a form titled 'Rule details' with the following fields and options:

- Rule name:** A text input field containing 'Require two approvals before merge'.
- Number of approvals needed:** A text input field containing the number '2'.
- Approval pool members - optional:** A section with explanatory text: '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.' Below this text is an 'Add' button.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Submit'.

7. (Opcional) Se deseja exigir que as aprovações de uma solicitação pull sejam provenientes de um grupo específico de usuários, em Approval rule members (Membros da regra de aprovação), escolha Add (Adicionar). Em Approver type (Tipo de aprovador), escolha uma das seguintes opções:
- Nome de usuário do IAM ou perfil assumido: essa opção preenche previamente o ID da conta da AWS com a conta que você usou para fazer login e requer apenas um nome. Ela pode ser usada para usuários de acesso federados e usuários do IAM cujos nomes correspondam ao nome fornecido. Essa é uma opção muito eficiente que oferece uma grande flexibilidade. Por exemplo, se você escolher essa opção, estiver conectado com a conta 123456789012 da Amazon Web Services e especificar **Mary_Major**, todas as opções a seguir contarão como aprovações provenientes desse usuário:
 - Um usuário do IAM na conta (`arn:aws:iam::123456789012:user/Mary_Major`)

- Um usuário federado identificado no IAM como Mary_Major
(arn:aws:sts::123456789012:federated-user/Mary_Major)

Essa opção não reconhecerá uma sessão ativa de alguém que assume a função de **CodeCommitReview** com um nome de sessão de função de Mary_Major (arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major), a menos que você inclua um curinga (*Mary_Major). Você também pode especificar o nome da função explicitamente (CodeCommitReview/Mary_Major).

- ARN totalmente qualificado: essa opção permite especificar o nome do recurso da Amazon (ARN) totalmente qualificado do perfil ou usuário do IAM. Essa opção também oferece suporte a funções assumidas usadas por outros AWS serviços, como AWS Lambda AWS CodeBuild e. Para funções assumidas, o formato do ARN deve ser arn:aws:sts::*AccountID*:assumed-role/*RoleName* para funções (roles) e arn:aws:sts::*AccountID*:assumed-role/*FunctionName* funções (functions).

Se você escolheu o nome de usuário do IAM ou perfil assumido como o tipo de aprovador, em Valor, insira o nome do usuário ou perfil do IAM ou o ARN totalmente qualificado do usuário ou perfil. Escolha Add (Adicionar) novamente para adicionar mais usuários ou funções, até que você tenha adicionado todos os usuários ou funções cujas aprovações contam para o número de aprovações necessárias.

Ambos os tipos de aprovadores permitem usar curingas (*) em seus valores. Por exemplo, se você escolher a opção nome de usuário do IAM ou perfil assumido e especificar **CodeCommitReview/***, todos os usuários que assumirem o perfil de **CodeCommitReview** serão contados no grupo de aprovação. Os nomes de sessão de função individual são contabilizados no número necessário de aprovadores. Dessa forma, tanto Mary_Major quanto Li_Juan contarão como aprovações quando conectados e assumindo a função de CodeCommitReview. Para obter mais informações sobre IAM ARNs, curingas e formatos, consulte [Identificadores do IAM](#).

 Note

As regras de aprovação não oferecem suporte a aprovações entre contas.

8. Quando terminar de configurar a regra de aprovação, escolha Submit (Enviar).

Criar uma regra de aprovação de uma solicitação pull (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para criar uma regra de aprovação para uma pull request em um CodeCommit repositório

- Execute o comando `create-pull-request-approval-rule`, especificando:
 - O ID da solicitação pull (com a opção `--id`).
 - O nome da regra de aprovação (com a opção `--approval-rule-name`).
 - O conteúdo da regra de aprovação (com a opção `--approval-rule-content`).

Ao criar a regra de aprovação, é possível especificar aprovadores em um grupo de aprovações de duas maneiras:

- **CodeCommitApprovers**: Essa opção requer somente uma conta e um recurso da Amazon Web Services. Ela pode ser usada para usuários de acesso federados e usuários do IAM cujo nome corresponde ao nome do recurso fornecido. Essa é uma opção muito eficiente que oferece uma grande flexibilidade. Por exemplo, se você especificar a conta 123456789012 da Amazon Web Services e **Mary_Major**, todas as opções a seguir contam como aprovações provenientes desse usuário:
 - Um usuário do IAM na conta (`arn:aws:iam::123456789012:user/Mary_Major`)
 - Um usuário federado identificado no IAM como `Mary_Major` (`arn:aws:sts::123456789012:federated-user/Mary_Major`)

Essa opção não reconhecerá uma sessão ativa de alguém que assume a função de **CodeCommitReview** com um nome de sessão de função de `Mary_Major` (`arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major`), a menos que você inclua um curinga (`*Mary_Major`).

- **ARN totalmente qualificado**: essa opção permite especificar o nome do recurso da Amazon (ARN) totalmente qualificado do perfil ou usuário do IAM.

Para obter mais informações sobre IAM ARNs, curingas e formatos, consulte [Identificadores do IAM](#).

O exemplo a seguir cria uma regra de aprovação chamada `Require two approved approvers` de uma solicitação pull com o ID 27. A regra especifica que duas aprovações são necessárias de um grupo de aprovações. O pool inclui todos os usuários que acessam CodeCommit e assumem a função **CodeCommitReview** na conta da 123456789012 Amazon Web Services. Ele também inclui um usuário federado ou usuário do IAM chamado `Nikhil_Jayashankar` na mesma conta da 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/*\"]}]}"
```

Exibir pull requests em um AWS CodeCommit repositório

Você pode usar o AWS CodeCommit console ou o AWS CLI para visualizar pull requests para seu repositório. Por padrão, você vê somente solicitações de extração em aberto, mas pode alterar o filtro para exibir todas as solicitações de extração, somente as solicitações fechadas, somente as solicitações de extração que você criou e mais.

Tópicos

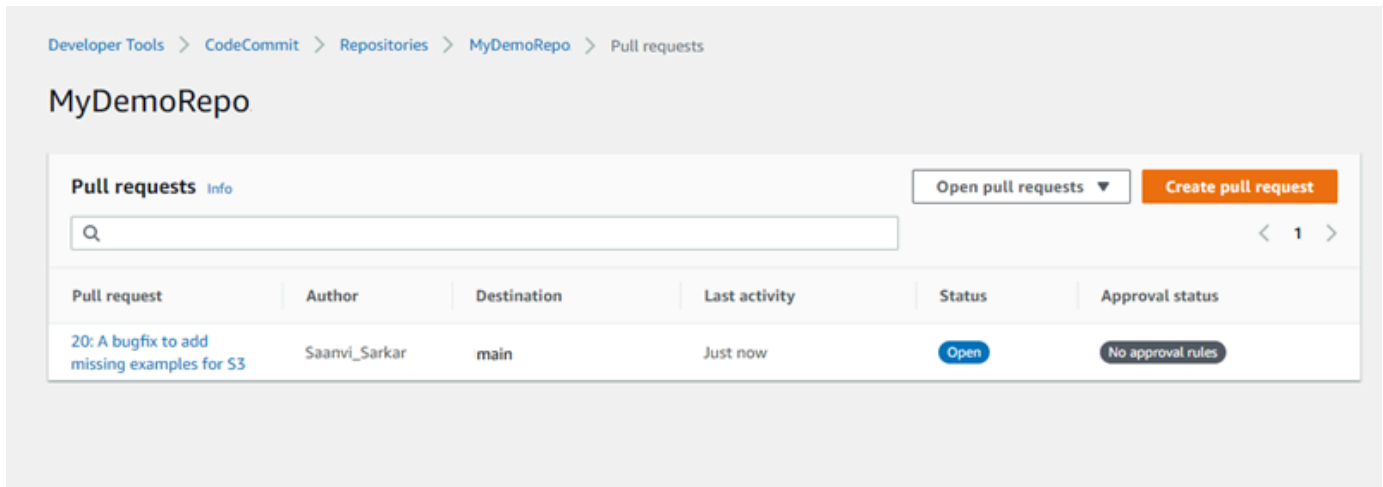
- [Visualização de solicitações pull \(console\)](#)
- [Visualizar solicitações pull \(AWS CLI\)](#)

Visualização de solicitações pull (console)

Você pode usar o AWS CodeCommit console para ver uma lista de pull requests em um CodeCommit repositório. Ao alterar o filtro, você pode alterar a exibição em lista para mostrar apenas um determinado conjunto de solicitações de extração. Por exemplo, você pode exibir uma lista de solicitações de extração que você criou com status de Open ou pode escolher um filtro diferente e exibir solicitações de extração que você criou com status de Closed.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja visualizar solicitações de extração.

3. No painel de navegação, escolha Pull Requests.
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas.



5. Para alterar o filtro de exibição, escolha na lista de filtros disponíveis:
 - All open requests (Todas as solicitações em aberto) (padrão): exibe todas as solicitações de extração com status de Open (Em aberto).
 - All pull requests (Todas as solicitações de extração): exibe todas as solicitações de extração.
 - All closed requests (Todas as solicitações fechadas): exibe todas as solicitações de extração com status de Closed (Fechadas).
 - My pull requests (Minhas solicitações de extração): exibe todas as solicitações de extração que você criou, independentemente do status. Ela não mostra revisões nas quais você comentou ou das quais participou.
 - My open requests (Minhas solicitações em aberto): exibe todas as solicitações de extração que você criou cujo status é Open (Aberto).
 - My closed requests (Minhas solicitações fechadas): exibe todas as solicitações de extração que você criou cujo status é Closed (Fechado).
6. Quando você encontrar uma solicitação pull na lista exibida que gostaria de exibir, escolha-a.

Visualizar solicitações pull (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Siga estas etapas para usar o AWS CLI para visualizar pull requests em um CodeCommit repositório.

1. Para visualizar uma lista de solicitações de extração em um repositório, execute o comando `list-pull-requests` especificando:
 - O nome do CodeCommit repositório em que você deseja visualizar pull requests (com a `--repository-name` opção).
 - (Opcional) O status da solicitação pull (com a opção `--pull-request-status`).
 - (Opcional) O nome do recurso da Amazon (ARN) do usuário do IAM que criou a solicitação pull (com a opção `--author-arn`).
 - (Opcional) Um token de enumeração que pode ser usado para retornar lotes de resultados (com a opção `--next-token`).
 - (Opcional) Um limite no número de resultados retornados por solicitação (com a opção `--max-results`).

Por exemplo, para listar pull requests criadas por um usuário do IAM com o ARN `arn:aws:iam::111111111111:user/Li_Juan` e o status de `CLOSED` em um CodeCommit repositório chamado: `MyDemoRepo`

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "nextToken": "",
  "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

IDs As solicitações pull são exibidas na ordem da atividade mais recente.

2. Para visualizar detalhes de uma solicitação pull específica, execute o comando `get-pull-request` com a opção `--pull-request-id` especificando o ID da solicitação pull. Por exemplo, para visualizar informações sobre uma pull request com o ID de `27`:

```
aws codecommit get-pull-request --pull-request-id 27
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\n\"Version\": \"2018-11-08\", \"Statements\n\": [{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\n\": [\n\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}}\",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 1571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "lastActivityDate": 1562619583.565,
    "pullRequestTargets": [
      {
        "sourceCommit": "ca45e279EXAMPLE",
        "sourceReference": "refs/heads/bugfix-1234",
        "mergeBase": "a99f5ddbEXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": false
        },
        "destinationCommit": "2abfc6beEXAMPLE",
        "repositoryName": "MyDemoRepo"
      }
    ],
    "revisionId": "e47def21EXAMPLE",
    "title": "Quick fix for bug 1234",
    "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
    "clientRequestToken": "d8d7612e-EXAMPLE",
    "creationDate": 1562619583.565,
    "pullRequestId": "27",
    "pullRequestStatus": "OPEN"
  }
}
```

3.

Para visualizar aprovações em uma solicitação pull, execute o comando `get-pull-request-approval-state`, especificando:

- O ID da solicitação pull (usando a opção `--pull-request-id`).
- O ID de revisão da solicitação pull (usando a `--revision-id` option). É possível obter o ID de revisão atual para uma solicitação pull usando o comando [get-pull-request](#).

Por exemplo, para ver as aprovações em uma pull request com um ID de **8** e um ID de revisão de **9f29d167EXAMPLE**:

```
aws codecommit get-pull-request-approval-state --pull-request-id 8 --revision-id 9f29d167EXAMPLE
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "approvals": [
    {
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
      "approvalState": "APPROVE"
    }
  ]
}
```

4. Para visualizar eventos de uma solicitação pull específica, execute o comando `describe-pull-request-events` com a opção `--pull-request-id` especificando o ID da solicitação pull. Por exemplo, para visualizar os eventos de uma pull request com o ID de **8**:

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "pullRequestEvents": [
    {
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_CREATED",
      "eventDate": 1510341779.53,
      "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
    },
    {
      "pullRequestStatusChangedEventMetadata": {
```

```

        "pullRequestStatus": "CLOSED"
      },
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
      "eventDate": 1510341930.72,
      "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
    }
  ]
}

```

5. Para visualizar se há conflitos de mesclagem para uma solicitação pull, execute o comando `get-merge-conflicts` especificando:

- O nome do CodeCommit repositório (com a `--repository-name` opção).
- A ramificação, tag, HEAD ou outra referência totalmente qualificada para a origem das alterações a serem usadas na avaliação da mesclagem (com a opção `--source-commit-specifier`).
- A ramificação, tag, HEAD ou outra referência totalmente qualificada para o destino das alterações a serem usadas na avaliação da mesclagem (com a opção `--destination-commit-specifier`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`)

Por exemplo, para ver se há algum conflito de mesclagem entre a ponta de uma ramificação de origem chamada *my-feature-branch* e uma ramificação de destino nomeada *main* em um repositório chamado: *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```

{
  "destinationCommitId": "fac04518EXAMPLE",
  "mergeable": false,
  "sourceCommitId": "16d097f03EXAMPLE"
}

```

Revisar uma solicitação pull

Você pode usar o AWS CodeCommit console para revisar as alterações incluídas em uma pull request. Você pode adicionar comentários à solicitação, aos arquivos e a linhas de código individuais. Você também pode responder aos comentários feitos por outros usuários. Se o seu repositório estiver [configurado com notificações](#), você receberá e-mails quando os usuários responderem aos seus comentários ou quando eles comentarem em uma solicitação pull.

Você pode usar o AWS CLI para comentar em uma pull request e responder aos comentários. Para revisar as alterações, você deve usar o CodeCommit console, o git diff comando ou uma ferramenta de comparação.

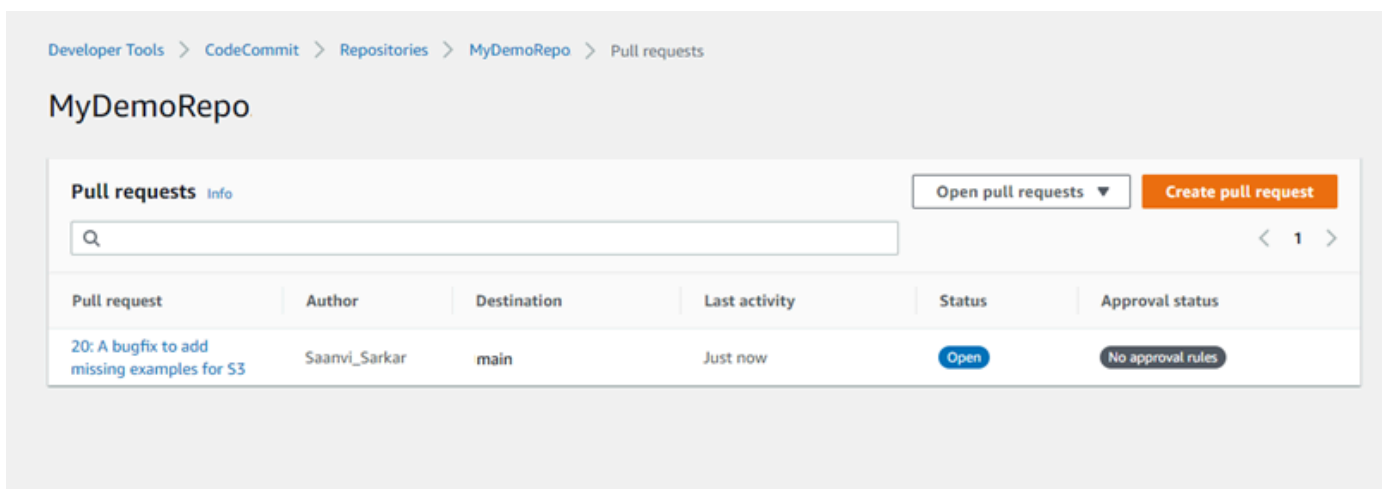
Tópicos

- [Revisar uma solicitação pull \(console\)](#)
- [Revisar solicitações pull \(AWS CLI\)](#)

Revisar uma solicitação pull (console)

Você pode usar o CodeCommit console para revisar uma pull request em um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório.
3. No painel de navegação, escolha Pull Requests (Solicitações pull).
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas. Escolha a solicitação pull aberta que você deseja analisar.



 Note

É possível comentar sobre uma solicitação pull fechada ou mesclada, mas não mesclá-la ou reabri-la.

5. Na solicitação pull, escolha Changes.
6. Execute um destes procedimentos:
 - Para adicionar um comentário geral para toda a solicitação pull, em Comments on changes (Comentários sobre alterações), escolha New comment (Novo comentário), insira um comentário e escolha Save (Salvar). Você pode usar [Markdown](#) ou inserir seu comentário em texto simples.



Comments on changes

New comment Preview markdown [Learn more](#)

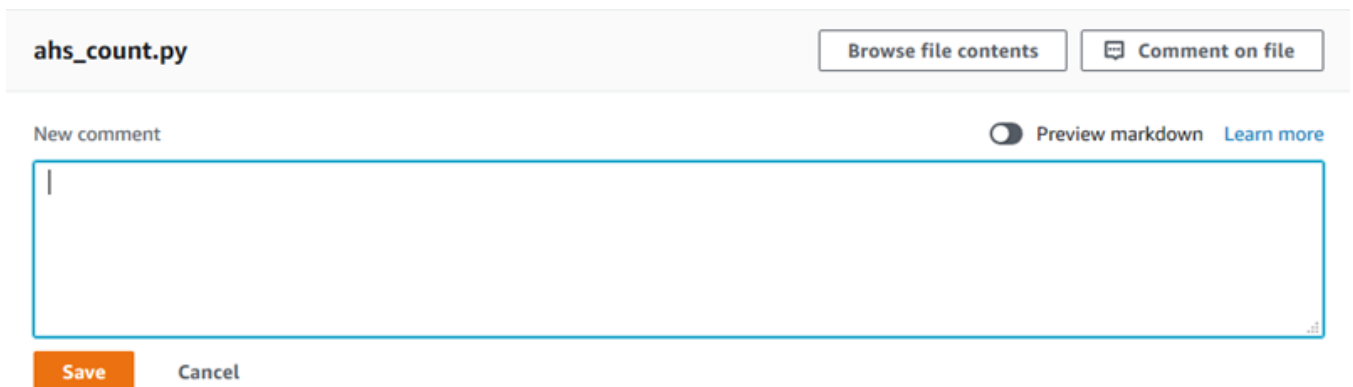
Did we also change the variable name in blf.py and concat.py?

Save

- Para adicionar um comentário a um arquivo na confirmação, em Changes, encontre o nome do arquivo. Escolha o ícone de comentário



que é exibido ao lado do nome do arquivo, insira um comentário e escolha Save (Salvar).



ahs_count.py Browse file contents Comment on file

New comment Preview markdown [Learn more](#)

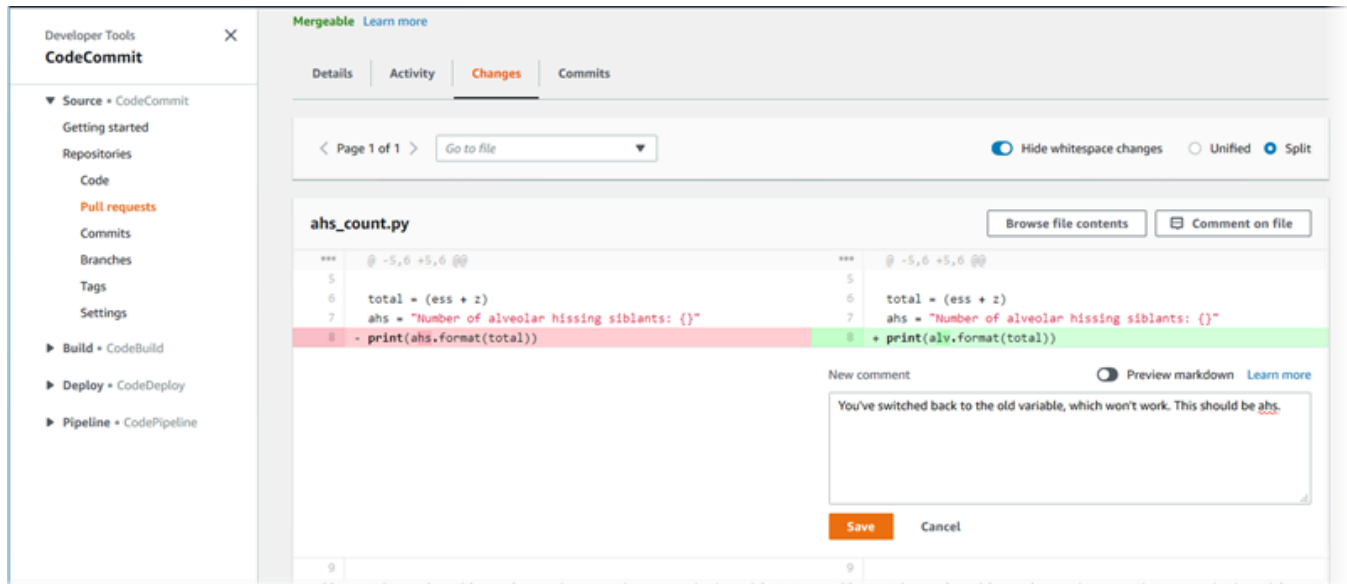
|

Save Cancel

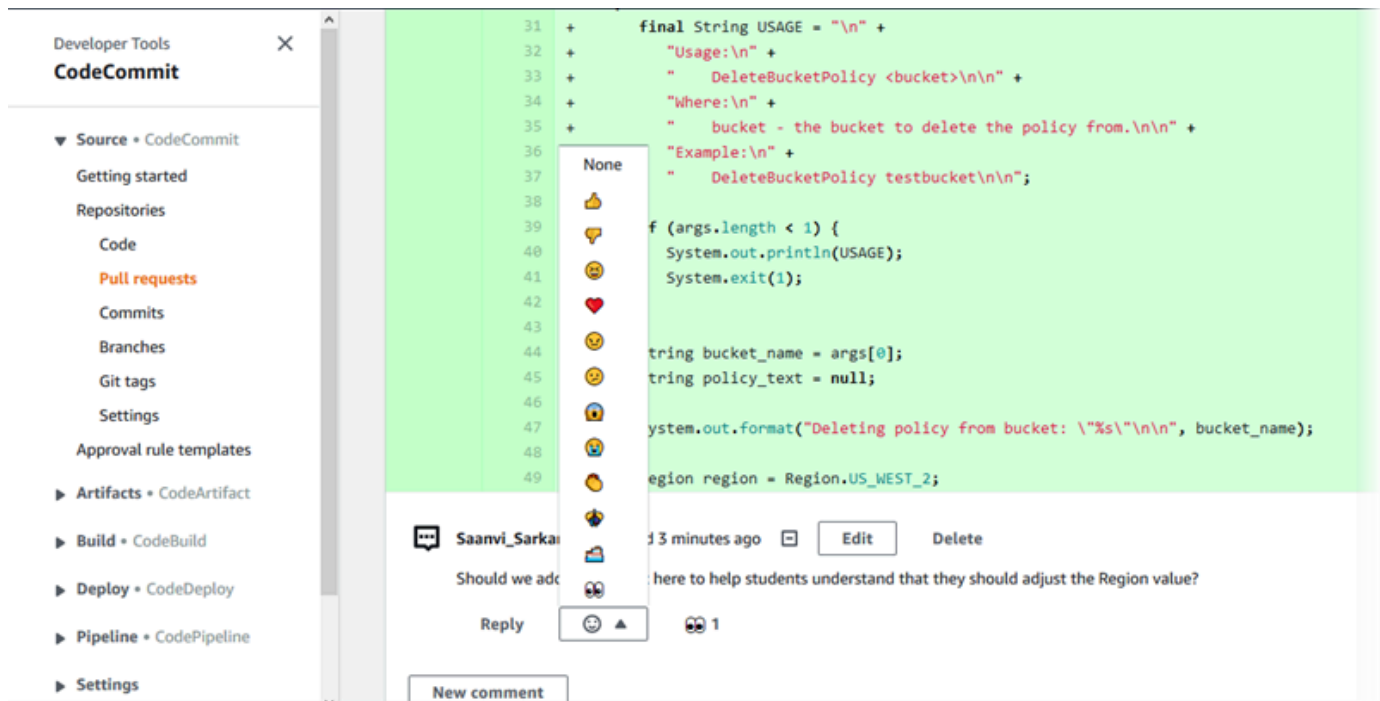
- Para adicionar um comentário a uma linha alterada na solicitação pull, em Changes, vá até a linha na qual deseja comentar. Escolha o ícone de comentário



que é exibido para essa linha, insira um comentário e escolha Save (Salvar).



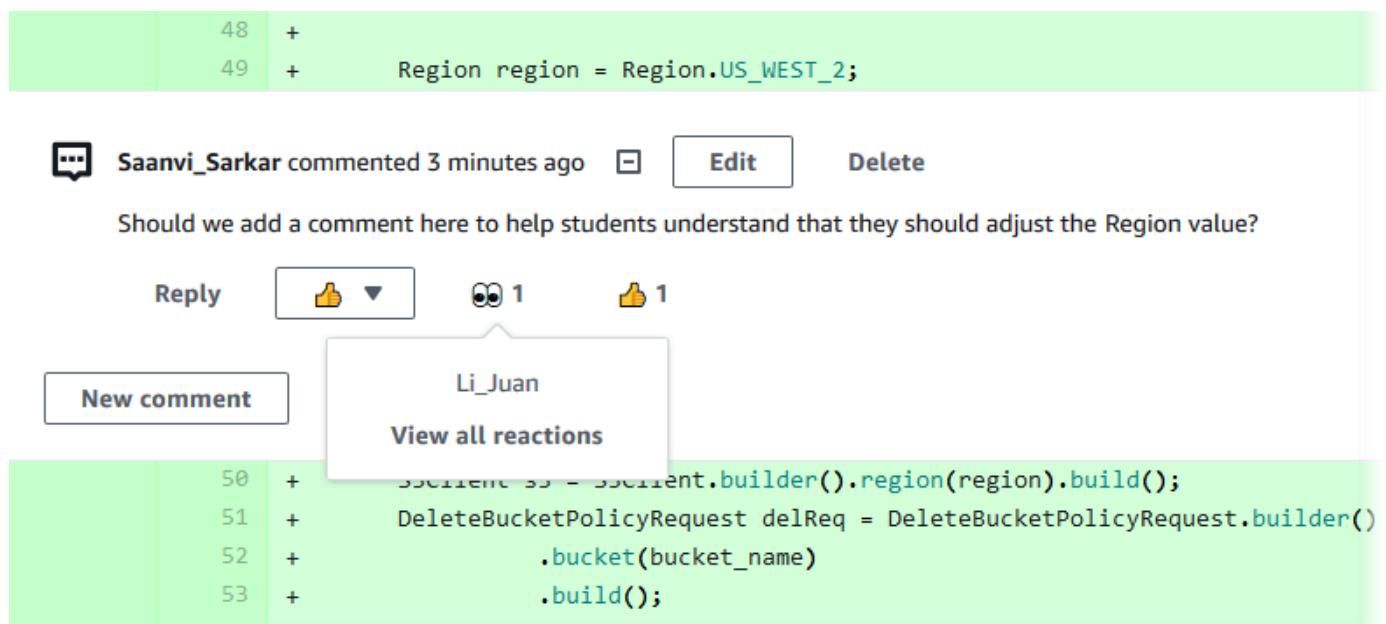
- Para responder a comentários em uma confirmação, em Changes ou Activity, escolha Reply. Você pode responder com texto e emojis.



Você pode ver os nomes das pessoas que responderam com uma reação de emoji específica selecionando essa reação. Para ver todas as reações de emojis e informações sobre quem respondeu com quais emojis, escolha Exibir todas as reações. Se você respondeu com um emoji a um comentário, sua resposta é mostrada no ícone do botão de reação de emoji.

 Note

As contagens de reações exibidas no console são precisas no momento em que a página foi carregada. Para obter as informações mais atuais sobre a contagem de reações de emojis, atualize a página ou selecione Exibir todas as reações.

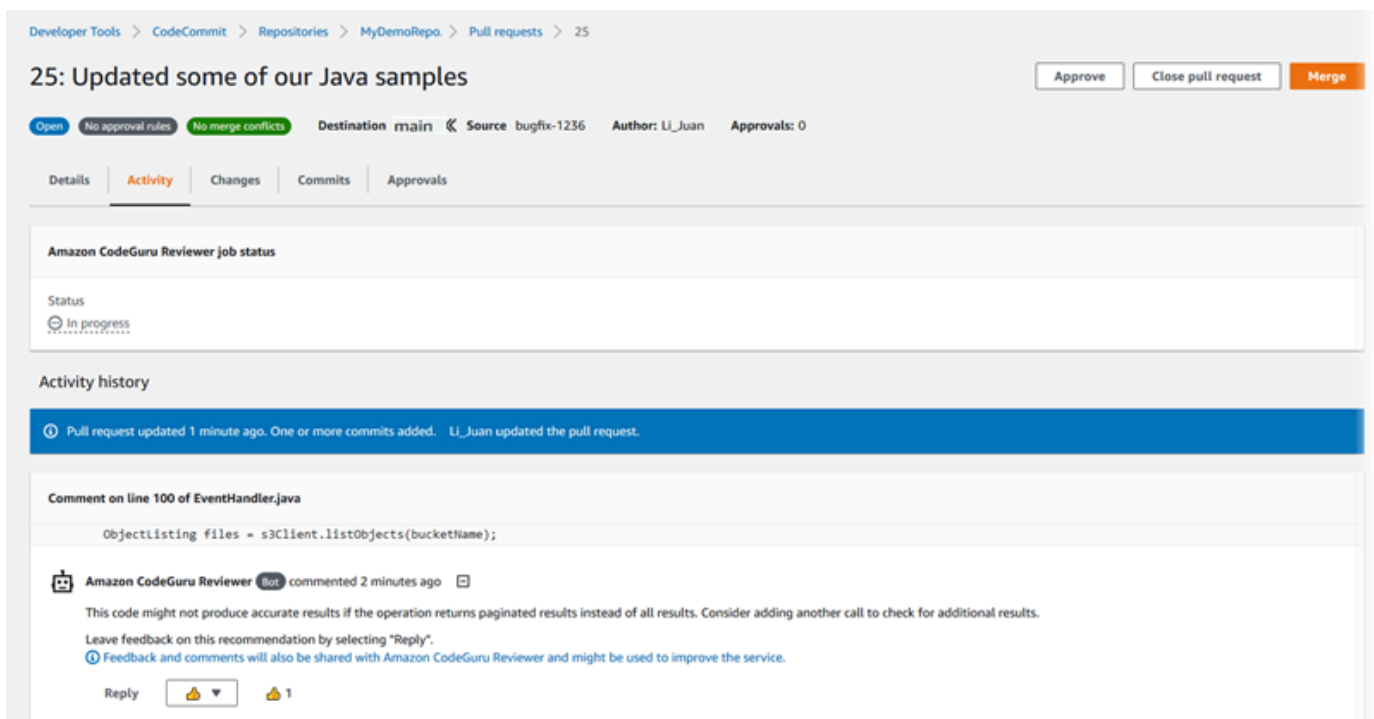


8. (Opcional) Para responder a uma recomendação criada pelo Amazon CodeGuru Reviewer, incluindo fornecer feedback sobre a qualidade da recomendação, escolha Responder. Use os botões de reação para fornecer informações gerais sobre se você aprova ou não a recomendação. Use o campo de comentário para fornecer mais detalhes sobre sua reação.

 Note

O Amazon CodeGuru Reviewer é um serviço automatizado de revisão de código que usa análise de programas e aprendizado de máquina para detectar problemas comuns e recomendar correções em seu código Java ou Python.

- Você só vê os comentários do Amazon CodeGuru Reviewer se tiver associado o repositório ao Amazon CodeGuru Reviewer, se a análise estiver completa e se o código na pull request for código Java ou Python. Para obter mais informações, consulte [Associar ou desassociar um AWS CodeCommit repositório com o Amazon Reviewer CodeGuru](#).
- Os comentários do Amazon CodeGuru Reviewer só aparecem na guia Alterações se tiverem sido feitos na revisão mais recente do pull request. Eles sempre aparecem na guia Atividade.
- Embora você possa responder com qualquer uma das reações de emoji disponíveis às recomendações do Amazon CodeGuru Reviewer, somente as reações de emoji de polegar para cima e de polegar para baixo são usadas para avaliar a utilidade da recomendação.



9. Para aprovar as alterações feitas em uma solicitação pull, escolha Approve (Aprovar).

Note

Não é possível aprovar uma solicitação pull que você criou.

É possível exibir aprovações, regras de aprovação e uma solicitação pull e regras de aprovação criadas por modelos de regra de aprovação em Approvals (Aprovações). Se você decidir que não deseja aprovar a solicitação pull, será possível escolher Revoke approval (Revogar aprovação).

Note

É possível aprovar ou revogar a aprovação somente em uma solicitação pull aberta. Não é possível aprovar ou revogar a aprovação em uma solicitação pull cujo status seja mesclado ou fechado.

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
Saarvi_Sarkar	Approved

Approval rules

Delete Edit Create approval rule

Approval rule	Status
My Approval Rule	Rule satisfied

Revisar solicitações pull (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Você pode revisar as pull requests com os seguintes AWS CLI comandos:

- [post-comment-for-pull-request](#), para adicionar comentários a uma solicitação pull
- [get-comments-for-pull-request](#), para ver os comentários deixados em uma solicitação pull

- [update-pull-request-approval-state](#), para aprovar ou revogar a aprovação de uma solicitação pull
- [post-comment-reply](#), para responder a um comentário em uma solicitação pull

Você também pode usar emojis com comentários em uma solicitação pull com os seguintes comandos:

- Para responder a um comentário com um emoji, execute [put-comment-reaction](#).
- Para ver as reações de emoji a um comentário, execute [get-comment-reactions](#).

Para usar o AWS CLI para revisar pull requests em um CodeCommit repositório

1. Para adicionar um comentário a uma solicitação pull em um repositório, execute o comando `post-comment-for-pull-request`, especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - O nome do repositório que contém a solicitação pull (com a opção `--repository-name`).
 - O ID de confirmação completo da confirmação na ramificação de destino onde a solicitação de extração é mesclada (com a opção `--before-commit-id`).
 - O ID de confirmação completo da confirmação na ramificação de origem que é a ponta atual da ramificação atual para a solicitação pull quando você publica o comentário (com a opção `--after-commit-id`).
 - Um token de idempotência exclusivo gerado pelo cliente (com a opção `--client-request-token`).
 - O conteúdo do seu comentário (com a opção `--content`).
 - Uma lista de informações de localização sobre onde colocar o comentário, incluindo:
 - O nome do arquivo que está sendo comparado, incluindo sua extensão e subdiretório, se houver (com o atributo `filePath`).
 - O número da linha da alteração em um arquivo comparado (com o atributo `filePosition`).
 - Se o comentário sobre a alteração estiver "antes" ou "depois" na comparação entre as ramificações de origem e destino (com o atributo `relativeFileVersion`).

Por exemplo, use esse comando para adicionar o comentário *"These don't appear to be used anywhere. Can we remove them?"* sobre a alteração no `ahs_count.py` arquivo em uma pull request com o ID de `47` em um repositório chamado `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 houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
  "afterBlobId": "1f330709EXAMPLE",
  "afterCommitId": "5d036259EXAMPLE",
  "beforeBlobId": "80906a4cEXAMPLE",
  "beforeCommitId": "317f8570EXAMPLE",
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
    "clientRequestToken": "123Example",
    "commentId": "abcd1234EXAMPLEb5678efgh",
    "content": "These don't appear to be used anywhere. Can we remove
them?",
    "creationDate": 1508369622.123,
    "deleted": false,
    "lastModifiedDate": 1508369622.123,
    "callerReactions": [],
    "reactionCounts": []
  },
  "location": {
    "filePath": "ahs_count.py",
    "filePosition": 367,
    "relativeFileVersion": "AFTER"
  },
  "repositoryName": "MyDemoRepo",
  "pullRequestId": "47"
}
```

2. Para exibir comentários de uma solicitação de pull, execute o comando `get-comments-for-pull-request`, especificando:

- O nome do CodeCommit repositório (com a `--repository-name` opção).
- O ID gerado pelo sistema da solicitação pull (com a opção `--pull-request-id`).
- (Opcional) Um token de enumeração para retornar o lote seguintes de resultados (com a opção `--next-token`).

- (Opcional) Um inteiro não negativo no número de resultados retornados (com a opção `--max-results`).

Por exemplo, use esse comando para visualizar comentários de uma solicitação pull com um ID 42.

```
aws codecommit get-comments-for-pull-request --pull-request-id 42
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
  "commentsForPullRequestData": [
    {
      "afterBlobId": "1f330709EXAMPLE",
      "afterCommitId": "5d036259EXAMPLE",
      "beforeBlobId": "80906a4cEXAMPLE",
      "beforeCommitId": "317f8570EXAMPLE",
      "comments": [
        {
          "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
          "clientRequestToken": "",
          "commentId": "abcd1234EXAMPLEb5678efgh",
          "content": "These don't appear to be used anywhere. Can we remove
them?",
          "creationDate": 1508369622.123,
          "deleted": false,
          "lastModifiedDate": 1508369622.123,
          "callerReactions": [],
          "reactionCounts":
          {
            "THUMBSUP" : 6,
            "CONFUSED" : 1
          }
        },
        {
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
          "clientRequestToken": "",
          "commentId": "442b498bEXAMPLE5756813",
          "content": "Good catch. I'll remove them.",
          "creationDate": 1508369829.104,
          "deleted": false,
```

```

        "lastModifiedDate": 150836912.273,
        "callerReactions": ["THUMBSUP"]
        "reactionCounts":
        {
            "THUMBSUP" : 14
        }
    },
    "location": {
        "filePath": "ahs_count.py",
        "filePosition": 367,
        "relativeFileVersion": "AFTER"
    },
    "repositoryName": "MyDemoRepo",
    "pullRequestId": "42"
}
],
"nextToken": "exampleToken"
}

```

3.

Para aprovar ou revogar a aprovação de uma solicitação pull, execute o comando `update-pull-request-approval-state`, especificando:

- O ID da solicitação pull (usando a opção `--pull-request-id`).
- O ID de revisão da solicitação pull (usando a `--revision-id` option). É possível obter o ID de revisão atual para uma solicitação pull usando o comando [get-pull-request](#).
- O estado de aprovação que você deseja aplicar (usando a opção `--approval-state`). Os estados de aprovação válidos incluem APPROVE e REVOKE.

Por exemplo, use esse comando para aprovar uma pull request com o ID de **27** e um ID de revisão de **9f29d167EXAMPLE**.

```
aws codecommit update-pull-request-approval-state --pull-request-id 27 --revision-id 9f29d167EXAMPLE --approval-state "APPROVE"
```

Se houver êxito, o comando não retornará nada.

4. Para enviar uma resposta a um comentário em uma solicitação pull, execute o comando `post-comment-reply`, especificando:

- O ID gerado pelo sistema do comentário ao qual você deseja responder (com a opção `--in-reply-to`).
- Um token de idempotência exclusivo gerado pelo cliente (com a opção `--client-request-token`).
- O conteúdo da sua resposta (com a opção `--content`).

Por exemplo, use esse comando para adicionar a resposta *"Good catch. I'll remove them."* ao comentário com o ID gerado pelo sistema de. *abcd1234EXAMPLEb5678efgh*

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --
content "Good catch. I'll remove them." --client-request-token 123Example
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
  "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": []
  }
}
```

Atualizar uma solicitação pull

Você pode atualizar uma solicitação pull com mais alterações no código enviando confirmações para a ramificação de origem de uma solicitação pull aberta. Para obter mais informações, consulte [Crie um commit em AWS CodeCommit](#).

Você pode usar o console do AWS CodeCommit ou o AWS CLI para atualizar o título ou a descrição de uma solicitação de extração. Você pode querer atualizar a solicitação pull ou descrição porque:

- Outros usuários não entendem a descrição ou o título original é enganador.

- Você quer que o título ou a descrição reflita as alterações feitas à ramificação de origem de uma solicitação pull aberta.

Atualizar uma solicitação pull (console)

Você pode usar o console do CodeCommit para atualizar o título e a descrição de uma solicitação pull em um repositório do CodeCommit. Você pode atualizar o código na solicitação pull, enviar via push confirmações para a ramificação de origem de uma solicitação pull aberta.

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja atualizar uma solicitação pull.
3. No painel de navegação, escolha Pull Requests (Solicitações pull).
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas. Escolha a solicitação pull aberta que você deseja atualizar.
5. Na solicitação pull, selecione Details (Detalhes) e, em seguida, escolha Edit details (Editar detalhes) para editar o título ou a descrição.

Note

Você não pode atualizar o título ou a descrição de uma solicitação pull fechada ou mesclada.

Atualizar solicitações pull (AWS CLI)

Para usar comandos da AWS CLI com o CodeCommit, instale a AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Você também pode estar interessado nos seguintes comandos:

- [update-pull-request-approval-state](#), para aprovar ou revogar a aprovação em uma solicitação pull.
- [create-pull-request-approval-rule](#), para criar uma regra de aprovação para uma solicitação pull.
- [delete-pull-request-approval-rule](#), para excluir uma regra de aprovação para uma solicitação pull.

- [Crie um commit usando o AWS CLI](#) ou [Criar uma confirmação usando um cliente Git](#), para criar e enviar via push alterações de código adicionais para a ramificação de origem de uma solicitação pull aberta.

Para usar a AWS CLI para atualizar solicitações pull em um repositório do CodeCommit

1. Para atualizar o título de uma solicitação pull em um repositório, execute o comando `update-pull-request-title` especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - O título da solicitação pull (com a opção `--title`).

Por exemplo, para atualizar o título de uma solicitação pull com o ID **47**:

```
aws codecommit update-pull-request-title --pull-request-id 47 --title
"Consolidation of global variables - updated review"
```

2. Para atualizar a descrição de uma solicitação pull, execute o comando `update-pull-request-description` especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - A descrição (com a opção `--description`).

Por exemplo, para atualizar a descrição de uma solicitação pull com o ID **47**:

```
aws codecommit update-pull-request-description --pull-request-id 47 --description
"Updated the pull request to remove unused global variable."
```

Editar ou excluir uma regra de aprovação de uma solicitação pull

Quando você tem uma regra de aprovação em uma solicitação pull, não é possível mesclar essa solicitação pull até que as condições sejam atendidas. É possível alterar as regras de aprovação de solicitações pull para facilitar o atendimento de suas condições ou aumentar o rigor das revisões. Você pode alterar o número de usuários que devem aprovar uma solicitação pull. Também é possível adicionar, remover ou alterar a associação em um grupo de aprovações de usuários para a regra.

Por fim, se não quiser mais usar uma regra de aprovação de uma solicitação pull, você poderá excluí-la.

Note

Também é possível substituir regras de aprovação de uma solicitação pull. Para obter mais informações, consulte [Substituir regras de aprovação em uma solicitação pull](#).

Você pode usar o AWS CodeCommit console ou o AWS CLI para editar e excluir as regras de aprovação do seu repositório.

Tópicos

- [Editar ou excluir uma regra de aprovação de uma solicitação pull \(console\)](#)
- [Editar ou excluir uma regra de aprovação de uma solicitação pull \(AWS CLI\)](#)

Editar ou excluir uma regra de aprovação de uma solicitação pull (console)

Você pode usar o CodeCommit console para editar ou excluir uma regra de aprovação para uma pull request em um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório no qual você deseja editar ou excluir uma regra de aprovação de uma solicitação pull.
3. No painel de navegação, escolha Pull Requests.
4. Escolha a solicitação pull em que você deseja editar ou excluir uma regra de aprovação. É possível editar e excluir regras de aprovação somente de solicitações pull abertas.

5. Na solicitação pull, escolha Approvals (Aprovações) e selecione a regra que deseja editar ou excluir da lista. Execute um destes procedimentos:
 - Se quiser editar a regra, escolha Edit (Editar).
 - Se desejar excluir a regra, escolha Delete (Excluir) e siga as instruções para verificar a exclusão da regra.
6. Em Edit approval rule (Editar regra de aprovação), faça as alterações desejadas na regra e escolha Submit (Enviar).

Edit approval rule

Rule details

Rule name
My Approval Rule

Number of approvals needed

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

7. Quando terminar de configurar a regra de aprovação, escolha Submit (Enviar).

Editar ou excluir uma regra de aprovação de uma solicitação pull (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Você pode usar o AWS CLI para editar o conteúdo de uma regra de aprovação e excluir uma regra de aprovação.

Note

Você também pode estar interessado nos seguintes comandos:

- [update-pull-request-approval-state](#), para aprovar ou revogar a aprovação em uma solicitação pull.
- [get-pull-request-approval-states](#), para visualizar as aprovações na solicitação pull.
- [evaluate-pull-request-approval-rules](#), para determinar se as regras de aprovação de uma solicitação pull tiveram suas condições atendidas.

Para usar o AWS CLI para editar ou excluir uma regra de aprovação para uma pull request em um CodeCommit repositório

1. Para editar uma regra de aprovação, execute o comando `update-pull-request-approval-rule-content`, especificando:
 - O ID da solicitação pull (com a opção `--id`).
 - O nome da regra de aprovação (com a opção `--approval-rule-name`).
 - O conteúdo da regra de aprovação (com a opção `--approval-rule-content`).

Este exemplo atualiza uma regra de aprovação nomeada *Require two approved approvers* para uma pull request com o ID de *27*. A regra exige a aprovação de um usuário de um grupo de aprovação que inclui qualquer usuário do IAM na conta da *123456789012* Amazon Web Services:

```
aws codecommit update-pull-request-approval-rule-content --pull-request-id 27
--approval-rule-name "Require two approved approvers" --approval-rule-content
```

```
"{Version: 2018-11-08, Statements: [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:[\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

2.

Para excluir uma regra de aprovação, execute o comando `delete-pull-request-approval-rule`, especificando:

- O ID da solicitação pull (com a opção `--id`).
- O nome da regra de aprovação (com a opção `--approval-rule-name`).

Por exemplo, para excluir uma regra de aprovação com o nome *My Approval Rule* de uma pull request com o ID de *15*:

```
aws codecommit delete-pull-request-approval-rule --pull-request-id 15 --approval-rule-name "My Approval Rule"
```

Substituir regras de aprovação em uma solicitação pull

No curso normal de desenvolvimento, convém que os usuários atendam às condições de regras de aprovação antes de mesclar solicitações pull. No entanto, pode haver momentos em que é necessário agilizar a mesclagem de uma solicitação pull. Por exemplo, é possível que você queira colocar uma correção de bug em produção, mas ninguém no grupo de aprovações esteja disponível para aprovar a solicitação pull. Em casos como esses, é possível optar por substituir as regras de aprovação em uma solicitação pull. Você pode substituir todas as regras de aprovação de uma solicitação pull, incluindo aquelas criadas especificamente para a solicitação pull e geradas a partir de um modelo de regra de aprovação. Não é possível substituir seletivamente uma regra de aprovação específica, somente todas as regras. Depois de reservar os requisitos de regra de aprovação substituindo as regras, é possível mesclar a solicitação pull com a ramificação de destino.

Ao substituir regras de aprovação em uma solicitação pull, as informações sobre o usuário que substituiu as regras são registradas na atividade da solicitação pull. Dessa forma, é possível voltar ao histórico de uma solicitação pull e examinar quem substituiu as regras. Você também poderá optar por revogar a substituição se a solicitação pull ainda estiver aberta. Depois que a solicitação pull for mesclada, você não poderá mais revogar a substituição.

Tópicos

- [Substituir regras de aprovação \(console\)](#)

- [Substituir regras de aprovação \(AWS CLI\)](#)

Substituir regras de aprovação (console)

É possível substituir os requisitos de regras de aprovação em uma solicitação pull no console, como parte da revisão de uma solicitação pull. Se você mudar de ideia, poderá revogar a substituição, e os requisitos da regra de aprovação serão reaplicados. Você só poderá substituir regras de aprovação ou revogar uma substituição se a solicitação pull ainda estiver aberta. Se ela for mesclada ou fechada, não será possível alterar seu estado de substituição.

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório.
3. No painel de navegação, escolha Pull Requests (Solicitações pull). Escolha a solicitação pull na qual deseja substituir os requisitos de regra de aprovação ou revogar uma substituição.
4. Na guia Approvals (Aprovações), escolha Override approval rules (Substituir regras de aprovação). Os requisitos são reservados, e o texto do botão é alterado para Revoke override (Revogar substituição). Para reaplicar os requisitos da regra de aprovação, escolha Revoke override (Revogar substituição).

Substituir regras de aprovação (AWS CLI)

É possível usar a AWS CLI para substituir os requisitos de regra de aprovação. Também é possível usá-la para visualizar o status de substituição de uma solicitação pull.

Como substituir requisitos de regra de aprovação em uma solicitação pull

1. Em um terminal ou uma linha de comando, execute o comando `override-pull-request-approval-rules`, especificando:
 - O ID gerado pelo sistema da solicitação pull.
 - O ID de revisão mais recente da solicitação pull. Para visualizar essas informações, use `get-pull-request`.
 - O status que você deseja para a substituição, `OVERRIDE` ou `REVOKE`. O status `REVOKE` remove o status `OVERRIDE`, mas não é salvo.

Por exemplo, para substituir regras de aprovação em uma solicitação pull com um ID **34** e um ID de revisão **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8EXAMPLE --override-status OVERRIDE
```

2. Se houver êxito, o comando não retornará nada.
3. Para revogar a substituição em uma solicitação pull com um ID **34** e um ID de revisão **927df8d8EXAMPLE**:

```
aws codecommit override-pull-request-approval-rules --pull-request-id 34 --  
revision-id 927df8d8EXAMPLE --override-status REVOKE
```

Como obter informações sobre o status de substituição de uma solicitação pull

1. Em um terminal ou uma linha de comando, execute o comando `get-pull-request-override-state`, especificando:
 - O ID gerado pelo sistema da solicitação pull.
 - O ID de revisão mais recente da solicitação pull. Para visualizar essas informações, use `get-pull-request`.

Por exemplo, para visualizar o estado de substituição de uma solicitação pull com um ID **34** e um ID de revisão **927df8d8EXAMPLE**:

```
aws codecommit get-pull-request-override-state --pull-request-id 34 --revision-  
id 927df8d8EXAMPLE
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "overridden": true,  
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

Mesclar uma solicitação pull em um repositório do AWS CodeCommit

Depois que seu código for revisado e todas as regras de aprovação (se houver) na solicitação pull forem atendidas, será possível mesclar uma solicitação pull de várias maneiras:

- É possível usar o console para mesclar a ramificação de origem com a ramificação de destino usando uma das estratégias de mesclagem disponíveis, que também fecha a solicitação pull. Também é possível resolver conflitos de mesclagem no console. O console exibe uma mensagem que indica se a solicitação pull pode ser mesclada ou se os conflitos devem ser resolvidos. Quando todos os conflitos forem resolvidos e você escolher Merge (Mesclar), a mesclagem será realizada usando a estratégia de mesclagem escolhida. Avanço rápido é a estratégia de mesclagem padrão, que é a opção padrão para o Git. Dependendo do estado do código nas ramificações de origem e de destino, essa estratégia pode não estar disponível, mas outras opções podem estar, como de compressão ou três vias.
- É possível usar a AWS CLI para tentar mesclar e fechar a solicitação pull usando a estratégia de mesclagem de avanço rápido, compressão ou três vias.
- No seu computador local, você pode usar o comando `git merge` para mesclar a ramificação de origem na ramificação de destino e, em seguida, envie seu código mesclado por push para a ramificação de destino. Essa abordagem tem desvantagens que devem ser consideradas cuidadosamente. Ela mescla a solicitação pull independentemente de os requisitos para regras de aprovação na solicitação pull serem atendidos, contornando esses controles. Mesclar e enviar a ramificação de destino também fechará a solicitação pull automaticamente se a solicitação pull for mesclada usando a estratégia de mesclagem de avanço rápido. Uma vantagem dessa abordagem é que o comando `git merge` permite escolher opções ou estratégias de mesclagem que não estão disponíveis no console do CodeCommit. Para obter mais informações sobre opções de mesclagem e `git merge`, consulte [git-merge](#) ou a documentação do Git.

O CodeCommit fechará uma solicitação pull automaticamente se a ramificação de origem ou a ramificação de destino da solicitação pull for excluída.

Tópicos

- [Mesclar uma solicitação pull \(console\)](#)
- [Mesclar uma solicitação pull \(AWS CLI\)](#)

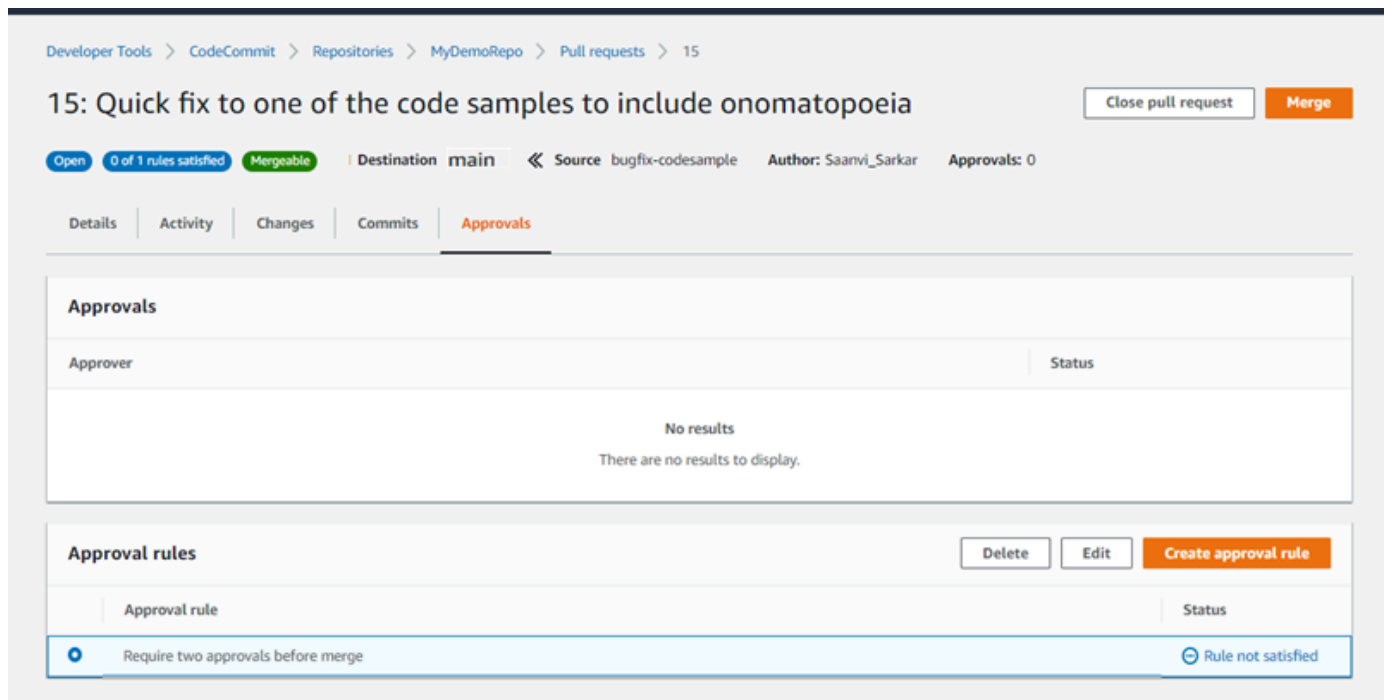
Mesclar uma solicitação pull (console)

Você pode usar o console do CodeCommit para mesclar uma solicitação pull em um repositório do CodeCommit. Depois que o status de uma solicitação pull for alterado para Merged (Mesclada), ela não será mais exibida na lista de solicitações pull abertas. Uma solicitação pull mesclada é categorizada como fechada. Ela não pode ser alterada novamente para Open (Aberta), mas os usuários ainda podem comentar sobre as alterações e responder aos comentários. Depois que uma solicitação pull for mesclada ou fechada, não será possível aprová-la, revogá-la ou substituir as regras de aprovação aplicadas à solicitação pull.

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório.
3. No painel de navegação, escolha Pull Requests (Solicitações pull).
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas. Escolha a solicitação pull aberta que você deseja mesclar.
5. Na solicitação pull, escolha Approvals (Aprovações). Revise a lista de aprovadores e verifique se todas as regras de aprovação (se houver) tiveram suas condições atendidas. Você não poderá mesclar uma solicitação pull se uma ou mais regras de aprovação tiverem o status Rule not satisfied (Regra não satisfeita). Se ninguém tiver aprovado a solicitação pull, avalie se deseja mesclá-la ou aguardar aprovações.

Note

Se uma regra de aprovação tiver sido criada para uma solicitação pull, você poderá editá-la ou excluí-la para desbloquear a mesclagem. Se a regra de aprovação tiver sido criada com um modelo de regra de aprovação, você não poderá editá-la nem excluí-la. É possível optar somente por substituir os requisitos. Para obter mais informações, consulte [Substituir regras de aprovação em uma solicitação pull](#).



6. Escolha Merge (Mesclar).
7. Na solicitação pull, escolha entre as estratégias de mesclagem disponíveis. As estratégias de mesclagem que não puderem ser aplicadas serão exibidas esmaecidas. Se não houver estratégias de mesclagem disponíveis, você pode optar por resolver conflitos manualmente no console do CodeCommit ou resolvê-los localmente usando o cliente Git. Para obter mais informações, consulte [Resolver conflitos em uma solicitação pull em um repositório do AWS CodeCommit](#).

Merge pull request 9: Bug fix for unhandled exception

Merge request details

Pull request: #9 Bug fix for unhandled exception

Destination main **Source** bugfix-bug1234

Merge strategy [Info](#)
Determines the way in which the current pull request will be merged into the destination branch

☐ **Fast forward merge**
`git merge --ff-only`
Merges the branches and moves the destination branch pointer to the tip of the source branch. This is the default merge strategy in Git.


☒ **Squash and merge**
`git merge --squash`
Combine all commits from the source branch into a single merge commit in the destination branch.


☐ **3-way merge**
`git merge --no-ff`
Create a merge commit and adds individual source commits to the destination branch.


Commit message - optional Preview markdown

Squashed commit of the following

commit d49940ad
Author: Li Juan <li_juan@example.com>
Date: Tue May 07 2019 15:12:48 GMT-0700 (Pacific Daylight Time)

Fixing the bug reported in 1234.

Author name
Maria Garcia

Email address
maria_garcia@example.com

☒ Delete source branch bugfix-bug1234 after merging?

[Cancel](#) [Merge pull request](#)

- A mesclagem de avanço rápido move a referência da ramificação de destino para a confirmação mais recente da ramificação de origem. Esse é o comportamento padrão do Git sempre que possível. Nenhuma confirmação de mesclagem é criada, mas todo o histórico de confirmações na ramificação de origem é retido como se tivesse ocorrido na ramificação de destino. As mesclagens de avanço rápido não são exibidas como uma mesclagem de ramificação na exibição do visualizador de confirmações do histórico da ramificação de destino porque nenhuma confirmação de mesclagem foi criada. A ponta da ramificação de origem é avançada rapidamente para a ponta da ramificação de destino.

- Uma mesclagem de compressão cria uma confirmação que contém as alterações na ramificação de origem e aplica essa única confirmação de compressão à ramificação de destino. Por padrão, a mensagem de confirmação para essa confirmação de compressão contém todas as mensagens de confirmação das alterações na ramificação de origem. Nenhuma alteração no histórico de confirmações individuais da ramificação é retida. Isso pode ajudar a simplificar o histórico do repositório e, ao mesmo tempo, manter uma representação gráfica da mesclagem na exibição do visualizador de confirmação do histórico da ramificação de destino.
 - Uma mesclagem de três vias cria uma confirmação de mesclagem para a mesclagem na ramificação de destino, mas também retém confirmações individuais feitas na ramificação de origem como parte do histórico da ramificação de destino. Isso pode ajudar a manter um histórico completo das alterações em seu repositório.
8. Se você escolher a estratégia de mesclagem de compressão ou três vias, revise a mensagem de confirmação gerada automaticamente e modifique-a se deseja alterar as informações. Adicione seu nome e endereço de e-mail para o histórico de confirmações.
 9. (Opcional) Desmarque a opção para excluir a ramificação de origem como parte da mesclagem. O padrão é excluir a ramificação de origem quando uma solicitação pull é mesclada.
 10. Selecione Merge pull request (Mesclar solicitação pull) para concluir a mesclagem.

Mesclar uma solicitação pull (AWS CLI)

Para usar comandos da AWS CLI com o CodeCommit, instale a AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar a AWS CLI para mesclar solicitações pull em um repositório do CodeCommit

1. Para avaliar se uma solicitação pull teve todas as regras de aprovação satisfeitas e está pronta para ser mesclada, execute o comando `evaluate-pull-request-approval-rules`, especificando:
 - O ID da solicitação pull (usando a opção `--pull-request-id`).
 - O ID de revisão da solicitação pull (usando a `--revision-id` option). É possível obter o ID de revisão atual para uma solicitação pull usando o comando [get-pull-request](#).

Por exemplo, para avaliar o estado das regras de aprovação em uma solicitação pull com um ID **27** e um ID de revisão **9f29d167EXAMPLE**:

```
aws codecommit evaluate-pull-request-approval-rules --pull-request-id 27 --  
revision-id 9f29d167EXAMPLE
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

 Note

Essa saída indica que uma solicitação pull não pode ser mesclada porque os requisitos de uma regra de aprovação não foram atendidos. Para mesclar essa solicitação pull, é possível fazer com que os revisores a aprovem para atender às condições da regra. Dependendo de suas permissões e de como a regra foi criada, também será possível editar, substituir ou excluir a regra. Para ter mais informações, consulte [Revisar uma solicitação pull](#), [Substituir regras de aprovação em uma solicitação pull](#) e [Editar ou excluir uma regra de aprovação de uma solicitação pull](#).

2. Para mesclar e fechar uma solicitação pull usando a estratégia de mesclagem de avanço rápido, execute o comando `merge-pull-request-by-fast-forward`, especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - O ID de confirmação completo da extremidade da ramificação de origem (com a opção `--source-commit-id`).
 - O nome do repositório (com a opção `--repository-name`).

Por exemplo, para mesclar e fechar uma solicitação pull com a ID `47` e uma ID de confirmação de origem `99132ab0EXAMPLE` em um repositório denominado `MyDemoRepo`:

```
aws codecommit merge-pull-request-by-fast-forward --pull-request-id 47 --source-
commit-id 99132ab0EXAMPLE --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\\"Version\\": \"2018-11-08\\",\\"Statements
\\": [{\\"Type\\": \\"Approvers\\",\\"NumberOfApprovalsNeeded\\": 1,\\"ApprovalPoolMembers
\\": [\\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\\"]]}]",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "I want one approver for this pull request",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.142,
    "description": "Review the latest changes and updates to the global
variables",
    "lastActivityDate": 1508887223.155,
    "pullRequestId": "47",
    "pullRequestStatus": "CLOSED",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": true,
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
      }
    ],
    "title": "Consolidation of global variables"
  }
}
```

```
}
}
```

3. Para mesclar e fechar uma solicitação pull usando a estratégia de mesclagem de compressão, execute o comando `merge-pull-request-by-squash`, especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - O ID de confirmação completo da extremidade da ramificação de origem (com a opção `--source-commit-id`).
 - O nome do repositório (com a opção `--repository-name`).
 - O nível de detalhes de conflitos que você deseja usar (com a opção `--conflict-detail-level`). Se não especificado, o **FILE_LEVEL** padrão será usado.
 - A estratégia de resolução de conflitos que você deseja usar (com a opção `--conflict-resolution-strategy`). Se não especificado, isso define `NONE` como padrão, e os conflitos devem ser resolvidos manualmente.
 - A mensagem de confirmação a ser incluída (com a opção `--commit-message`).
 - O nome a ser usado para a confirmação (com a opção `--author-name`).
 - O endereço de e-mail a ser usado para a confirmação (com a opção `--email`).
 - Opção de manter pastas vazias (com a opção `--keep-empty-folders`).

O exemplo a seguir mescla e fecha uma solicitação pull com o ID **47** e um ID de confirmação de origem **99132ab0EXAMPLE** em um repositório denominado **MyDemoRepo**. Ele usa os detalhes de conflitos `LINE_LEVEL` e a estratégia de resolução de conflitos `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"
```

Se bem-sucedido, esse comando produz o mesmo tipo de saída que a mesclagem por avanço rápido, uma saída semelhante à seguinte:

```
{
  "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"
}
}

```

4. Para mesclar e fechar uma solicitação pull usando a estratégia de mesclagem de três vias, execute o comando `merge-pull-request-by-three-way`, especificando:

- O ID da solicitação pull (com a opção `--pull-request-id`).
- O ID de confirmação completo da extremidade da ramificação de origem (com a opção `--source-commit-id`).
- O nome do repositório (com a opção `--repository-name`).
- O nível de detalhes de conflitos que você deseja usar (com a opção `--conflict-detail-level`). Se não especificado, o **FILE_LEVEL** padrão será usado.
- A estratégia de resolução de conflitos que você deseja usar (com a opção `--conflict-resolution-strategy`). Se não especificado, isso define NONE como padrão, e os conflitos devem ser resolvidos manualmente.
- A mensagem de confirmação a ser incluída (com a opção `--commit-message`).
- O nome a ser usado para a confirmação (com a opção `--author-name`).
- O endereço de e-mail a ser usado para a confirmação (com a opção `--email`).
- Opção de manter pastas vazias (com a opção `--keep-empty-folders`).

O exemplo a seguir mescla e fecha uma solicitação pull com o ID **47** e um ID de confirmação de origem **99132ab0EXAMPLE** em um repositório denominado **MyDemoRepo**. Ele usa as opções padrão para detalhes de conflitos e estratégia de resolução de conflitos:

```
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"
```

Se bem-sucedido, esse comando produz o mesmo tipo de saída que a mesclagem por avanço rápido, semelhante ao seguinte:

```
{
  "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"
}

```

Resolver conflitos em uma solicitação pull em um repositório do AWS CodeCommit

Se a sua solicitação pull tem conflitos e não pode ser mesclada, você pode tentar resolvê-los de várias maneiras:

- No computador local, você pode usar o comando `git diff` para localizar os conflitos entre as duas ramificações e fazer alterações para resolvê-los. Você também pode usar uma ferramenta de diferenciação ou outros softwares para ajudar você a localizar e resolver as diferenças. Depois de resolvê-los de forma satisfatória, você pode enviar por push a ramificação de origem com as alterações que contêm os conflitos resolvidos, o que atualizará a solicitação pull. Para obter mais informações sobre `git diff` e `git difftool`, consulte a documentação do Git.
- No console, você pode selecionar **Resolve conflicts (Resolver conflitos)**. Isso abre um editor de texto simples que mostra conflitos de maneira semelhante ao comando `git diff`. Você pode revisar os conflitos manualmente em cada arquivo que os contém, fazer as alterações e, em seguida, atualizar a solicitação pull com as alterações.
- No AWS CLI, você pode usar a AWS CLI para obter informações sobre conflitos de mesclagem e criar uma confirmação de mesclagem sem referência para testar uma mesclagem.

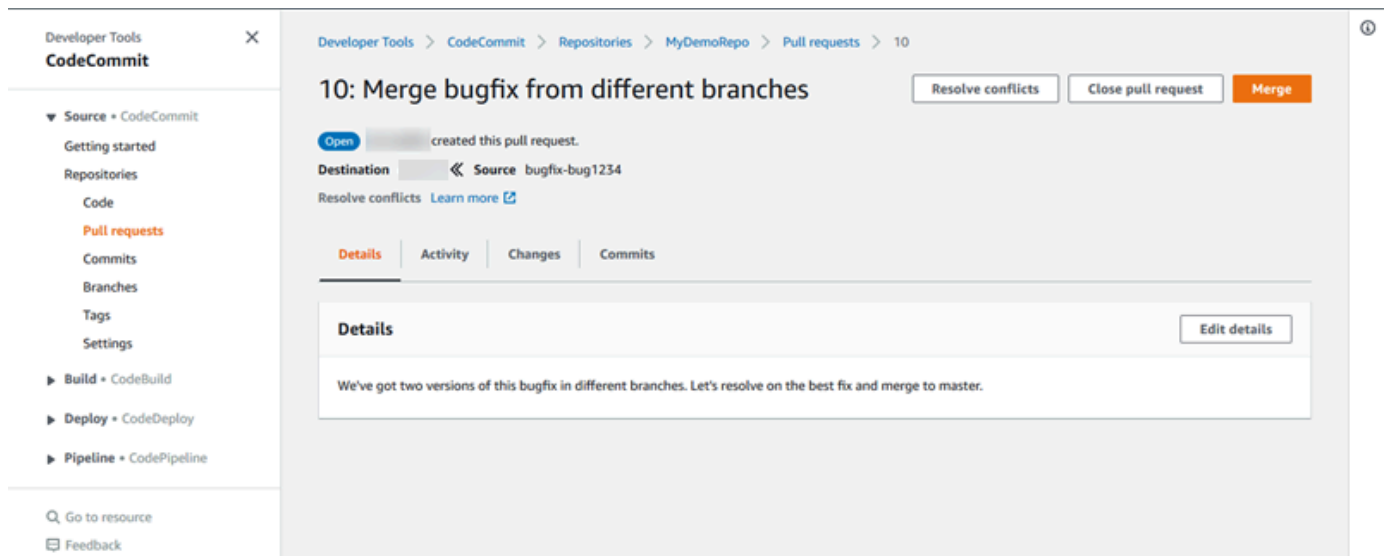
Tópicos

- [Resolver conflitos em uma solicitação pull \(console\)](#)
- [Resolver conflitos em uma solicitação pull \(AWS CLI\)](#)

Resolver conflitos em uma solicitação pull (console)

Você pode usar o console do CodeCommit para resolver conflitos em uma solicitação pull em um repositório do CodeCommit.

1. Abra o console do CodeCommit em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em **Repositories (Repositórios)**, escolha o nome do repositório.
3. No painel de navegação, escolha **Pull Requests (Solicitações pull)**.
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas. Escolha a solicitação pull aberta que você deseja mesclar, mas que contém conflitos.
5. Na solicitação pull, selecione **Resolve conflicts (Resolver conflitos)**. Essa opção só será exibida se houver conflitos que devem ser resolvidos para que a solicitação pull possa ser mesclada.



6. Uma janela de resolução de conflitos é aberta, listando cada arquivo que tem conflitos que devem ser resolvidos. Escolha cada arquivo na lista para revisar os conflitos e faça as alterações necessárias até que todos eles sejam resolvidos.

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

- Você pode optar por usar o conteúdo do arquivo de origem, o conteúdo do arquivo de destino, ou se o arquivo não é binário, para editar manualmente o conteúdo de um arquivo para que ele contenha apenas as alterações que você deseja. Os marcadores padrão git diff são usados para mostrar os conflitos entre o destino (HEAD) e as ramificações de origem no arquivo.
- Se um arquivo é binário, um submódulo do Git, ou se há um conflito de nome de arquivo/pasta, você deve optar por usar o arquivo de origem ou de destino para resolvê-los. Você não pode visualizar ou editar arquivos binários no console do CodeCommit.

- Se houver conflitos de modo de arquivo, você verá a opção para resolver esse conflito ao escolher entre o modo de arquivo do arquivo de origem e o modo de arquivo do arquivo de destino.
 - Se você decidir que deseja descartar as alterações em um arquivo e restaurá-lo para o estado que gerou conflito, selecione Reset file (Redefinir arquivo). Isso permite que você resolva os conflitos de forma diferente.
7. Quando você tiver certeza de suas alterações, selecione Update pull request (Atualizar solicitação pull).

 Note

Você deve resolver todos os conflitos em todos os arquivos para atualizar com êxito a solicitação pull com as alterações.

8. A solicitação pull será atualizada com as alterações e poderá ser mesclada. Você verá a página de mesclagem. Você pode optar por mesclar a solicitação pull neste momento ou pode retornar à lista de solicitações pull.

Resolver conflitos em uma solicitação pull (AWS CLI)

Para usar comandos da AWS CLI com o CodeCommit, instale a AWS CLI. Para obter mais informações, consulte [Referência da linha de comando](#).

Nenhum comando da AWS CLI sozinho permite que você resolva conflitos em uma solicitação pull e mescle essa solicitação. No entanto, você pode usar comandos individuais para descobrir conflitos, tentar resolvê-los e testar se uma solicitação pull pode ser mesclada. Você pode usar:

- `get-merge-options`, para descobrir as opções de mesclagem disponíveis para uma mesclagem entre dois especificadores de confirmação.
- `get-merge-conflicts`, para retornar uma lista de arquivos com conflitos de mesclagem em uma mesclagem entre dois especificadores de confirmação.
- `batch-describe-merge-conflicts`, para obter informações sobre todos os conflitos de mesclagem em arquivos de uma mesclagem entre duas confirmações usando uma estratégia de mesclagem especificada.
- `describe-merge-conflicts`, para obter informações detalhadas sobre conflitos de mesclagem de um arquivo específico entre duas confirmações usando uma estratégia de mesclagem especificada.

- `create-unreferenced-merge-commit`, para testar o resultado da mesclagem de dois especificadores de confirmação usando uma estratégia de mesclagem especificada.

1.

Para descobrir as opções de mesclagem disponíveis para uma mesclagem entre dois especificadores de confirmação, execute o comando `get-merge-options`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- O nome do repositório (com a opção `--repository-name`).
- (Opcional) Uma estratégia de resolução de conflitos a ser usada (com a opção `--conflict-resolution-strategy`).
- (Opcional) O nível de detalhes que deseja sobre os conflitos (com a opção `--conflict-detail-level`).

Por exemplo, para determinar as estratégias de mesclagem disponíveis para mesclar uma ramificação de origem chamada *bugfix-1234* com uma ramificação de destino denominada *principal* em um repositório denominado *MyDemoRepo*:

```
aws codecommit get-merge-options --source-commit-specifier bugfix-1234 --  
destination-commit-specifier main --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "mergeOptions": [  
    "FAST_FORWARD_MERGE",  
    "SQUASH_MERGE",  
    "THREE_WAY_MERGE"  
  ],  
  "sourceCommitId": "d49940adEXAMPLE",  
  "destinationCommitId": "86958e0aEXAMPLE",  
  "baseCommitId": "86958e0aEXAMPLE"  
}
```

2.

Para obter uma lista de arquivos que contêm conflitos de mesclagem em uma mesclagem entre dois especificadores de confirmação, execute o comando `get-merge-conflicts`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- O nome do repositório (com a opção `--repository-name`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`).
- (Opcional) O nível de detalhes que deseja sobre os conflitos (com a opção `--conflict-detail-level`).
- (Opcional) Uma estratégia de resolução de conflitos a ser usada (com a opção `--conflict-resolution-strategy`).
- (Opcional) O número máximo de arquivos com conflitos a serem retornados (com a opção `--max-conflict-files`).

Por exemplo, para obter uma lista de arquivos que contêm conflitos em uma mesclagem entre uma ramificação de origem denominada `feature-randomizationfeature` e uma ramificação de destino denominada `principal` usando a estratégia de mesclagem de três vias em um repositório denominado `MyDemoRepo`:

```
aws codecommit get-merge-conflicts --source-commit-specifier feature-
randomizationfeature --destination-commit-specifier main --merge-option
THREE_WAY_MERGE --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "mergeable": false,
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b6958EXAMPLE",
  "conflictMetadataList": [
    {
      "filePath": "readme.md",
      "fileSizes": {
        "source": 139,
```

```

        "destination": 230,
        "base": 85
    },
    "fileModes": {
        "source": "NORMAL",
        "destination": "NORMAL",
        "base": "NORMAL"
    },
    "objectTypes": {
        "source": "FILE",
        "destination": "FILE",
        "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
        "source": false,
        "destination": false,
        "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
        "source": "M",
        "destination": "M"
    }
}
]
}

```

3.

Para obter informações sobre conflitos de mesclagem em todos os arquivos ou um subconjunto de arquivos em uma mesclagem entre dois especificadores de confirmação, execute o comando `batch-describe-merge-conflicts`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`).
- O nome do repositório (com a opção `--repository-name`).

- (Opcional) Uma estratégia de resolução de conflitos a ser usada (com a opção `--conflict-resolution-strategy`).
- (Opcional) O nível de detalhes que deseja sobre os conflitos (com a opção `--conflict-detail-level`).
- (Opcional) O número máximo de blocos de mesclagem a serem retornados (com a opção `--max-merge-hunks`).
- (Opcional) O número máximo de arquivos com conflitos a serem retornados (com a opção `--max-conflict-files`).
- (Opcional) O caminho de arquivos de destino a ser usado para descrever os conflitos (com a opção `--file-paths`).

Por exemplo, para determinar os conflitos de mesclagem para mesclar uma ramificação de origem denominada *feature-randomizationfeature* com uma ramificação de destino denominada *principal* usando a estratégia *THREE_WAY_MERGE* em um repositório denominado *MyDemoRepo*:

```
aws codecommit batch-describe-merge-conflicts --source-commit-specifier feature-randomizationfeature --destination-commit-specifier main --merge-option THREE_WAY_MERGE --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "conflicts": [
    {
      "conflictMetadata": {
        "filePath": "readme.md",
        "fileSizes": {
          "source": 139,
          "destination": 230,
          "base": 85
        },
        "fileModes": {
          "source": "NORMAL",
          "destination": "NORMAL",
          "base": "NORMAL"
        },
        "objectTypes": {
```



```

        "source": "FILE",
        "destination": "FILE",
        "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
        "source": false,
        "destination": false,
        "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
        "source": "M",
        "destination": "M"
    }
},
"mergeHunks": [
    {
        "isConflict": true,
        "source": {
            "startLine": 0,
            "endLine": 3,
            "hunkContent": "VGhpcyBpEXAMPLE=="
        },
        "destination": {
            "startLine": 0,
            "endLine": 1,
            "hunkContent": "VXNlIHRoEXAMPLE="
        }
    }
]
},
"errors": [],
"destinationCommitId": "86958e0aEXAMPLE",
"sourceCommitId": "6ccd57fdEXAMPLE",
"baseCommitId": "767b6958EXAMPLE"
}

```

4.

Para obter informações detalhadas sobre conflitos de mesclagem de um arquivo específico em uma mesclagem entre dois especificadores de confirmação, execute o comando `describe-merge-conflicts`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`).
- O caminho do arquivo de destino a ser usado para descrever os conflitos (com a opção `--file-path`).
- O nome do repositório (com a opção `--repository-name`).
- (Opcional) Uma estratégia de resolução de conflitos a ser usada (com a opção `--conflict-resolution-strategy`).
- (Opcional) O nível de detalhes que deseja sobre os conflitos (com a opção `--conflict-detail-level`).
- (Opcional) O número máximo de blocos de mesclagem a serem retornados (com a opção `--max-merge-hunks`).
- (Opcional) O número máximo de arquivos com conflitos a serem retornados (com a opção `--max-conflict-files`).

Por exemplo, para determinar os conflitos de mesclagem em um arquivo denominado *readme.md* em uma ramificação de origem denominada *feature-randomizationfeature* com uma ramificação de destino chamada *principal* usando a estratégia *THREE_WAY_MERGE* em um repositório denominado *MyDemoRepo*:

```
aws codecommit describe-merge-conflicts --source-commit-specifier feature-randomizationfeature --destination-commit-specifier main --merge-option THREE_WAY_MERGE --file-path readme.md --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "conflictMetadata": {
    "filePath": "readme.md",
```

```

    "fileSizes": {
      "source": 139,
      "destination": 230,
      "base": 85
    },
    "fileModes": {
      "source": "NORMAL",
      "destination": "NORMAL",
      "base": "NORMAL"
    },
    "objectTypes": {
      "source": "FILE",
      "destination": "FILE",
      "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
      "source": false,
      "destination": false,
      "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  },
  "mergeHunks": [
    {
      "isConflict": true,
      "source": {
        "startLine": 0,
        "endLine": 3,
        "hunkContent": "VGhpcyBpEXAMPLE=="
      },
      "destination": {
        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ],

```

```
"destinationCommitId": "86958e0aEXAMPLE",  
"sourceCommitId": "6ccd57fdEXAMPLE",  
"baseCommitId": "767b69580EXAMPLE"  
}
```

5.

Para criar uma confirmação sem referência que representa o resultado da mesclagem de dois especificadores de confirmação, execute o comando `create-unreferenced-merge-commit`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`).
- O nome do repositório (com a opção `--repository-name`).
- (Opcional) Uma estratégia de resolução de conflitos a ser usada (com a opção `--conflict-resolution-strategy`).
- (Opcional) O nível de detalhes que deseja sobre os conflitos (com a opção `--conflict-detail-level`).
- (Opcional) A mensagem de confirmação a ser incluída (com a opção `--commit-message`).
- (Opcional) O nome a ser usado para a confirmação (com a opção `--name`).
- (Opcional) O endereço de e-mail a ser usado para a confirmação (com a opção `--email`).
- (Opcional) Opção de manter pastas vazias (com a opção `--keep-empty-folders`).

Por exemplo, para determinar os conflitos de mesclagem para mesclar uma ramificação de origem denominada *bugfix-1234* com uma ramificação de destino denominada *principal* usando a estratégia `ACCEPT_SOURCE` em um repositório chamado *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
"commitId": "4f178133EXAMPLE",  
"treeId": "389765daEXAMPLE"  
}
```

Fechar uma pull request em um AWS CodeCommit repositório

Se você deseja fechar uma solicitação pull sem mesclar o código, é possível fazer isso de várias maneiras:

- No console, você pode fechar uma solicitação de extração sem mesclar o código. Você pode querer fazer isso se quiser usar o comando `git merge` para mesclar as ramificações manualmente ou se o código na ramificação de origem da solicitação de extração não é um código que você deseja mesclar na ramificação de destino.
- Você pode excluir a ramificação de origem especificada na pull request. CodeCommit fecha uma pull request automaticamente se a ramificação de origem ou de destino da pull request for excluída.
- No AWS CLI, você pode atualizar o status de uma pull request de `OPEN` para `CLOSED`. Isso fecha a solicitação pull sem mesclar o código.

Tópicos

- [Fechar uma solicitação pull \(console\)](#)
- [Fechar uma solicitação pull \(AWS CLI\)](#)

Fechar uma solicitação pull (console)

Você pode usar o CodeCommit console para fechar uma pull request em um CodeCommit repositório. Depois de o status de uma solicitação de extração mudar para `Closed`, ele não poderá ser alterado de volta para `Open`, mas os usuários ainda poderão comentar as alterações e responder aos comentários.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em **Repositories** (Repositórios), escolha o nome do repositório.
3. No painel de navegação, escolha **Pull Requests** (Solicitações pull).
4. Por padrão, é exibida uma lista de todas as solicitações de extração abertas. Escolha a solicitação de extração aberta que você deseja fechar.

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. Na solicitação pull, selecione Close pull request (Fechar solicitação pull). Essa opção fecha a solicitação de pull sem tentar mesclar a ramificação de origem com a ramificação de destino. Essa opção não fornece uma maneira para excluir a ramificação de origem como parte do fechamento da solicitação pull, mas você pode fazer isso depois de fechar a solicitação.

Fechar uma solicitação pull (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para fechar pull requests em um CodeCommit repositório

- Para atualizar o status de uma solicitação de extração em um repositório de OPEN para CLOSED, execute o comando `update-pull-request-status`, especificando:
 - O ID da solicitação pull (com a opção `--pull-request-id`).
 - O status da solicitação de extração (com a opção `--pull-request-status`).

Por exemplo, para atualizar o status de uma pull request com o ID de **42** para um status de **CLOSED** em um CodeCommit repositório chamado `MyDemoRepo`:

```
aws codecommit update-pull-request-status --pull-request-id 42 --pull-request-status CLOSED
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codecommit:GetRepository",
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ]
    }
  ]
}
```

Trabalhar com modelos de regra de aprovação

É possível criar regras de aprovação de solicitações pull. Para aplicar regras de aprovação automaticamente a algumas ou a todas as solicitações pull criadas em repositórios, use modelos de regras de aprovação. Os modelos de regras de aprovação ajudam a personalizar seus fluxos de trabalho de desenvolvimento em vários repositórios para que diferentes ramificações tenham níveis apropriados de aprovações e controle. É possível definir regras diferentes para ramificações de produção e desenvolvimento. Essas regras serão aplicadas sempre que uma solicitação pull que corresponda às condições da regra for criada. Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Você pode associar um modelo de regra de aprovação a um ou mais repositórios no Região da AWS local em que eles foram criados. Quando um modelo é associado a um repositório, ele cria automaticamente regras de aprovação de solicitações pull nesse repositório como parte da criação da solicitação pull. Assim como uma regra de aprovação única, um modelo de regra de aprovação define uma estrutura de regra de aprovação, incluindo o número de aprovações necessárias e um grupo de usuários opcional dos quais as aprovações devem vir. Ao contrário de uma regra de aprovação, também é possível definir referências de destino (a ramificação ou as ramificações), também conhecidas como filtros de ramificação. Se você definir referências de destino, somente as solicitações pull cujos nomes de ramificação de destino correspondam aos nomes de ramificação especificados (referências de destino) no modelo terão regras criadas para elas. Portanto, por exemplo, se você especificar **refs/heads/main** como uma referência de destino, a regra de aprovação definida no modelo será aplicada somente a solicitações pull se a ramificação de destino for **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 ✕

Tópicos

- [Criar um modelo de regra de aprovação](#)
- [Associar um modelo de regra de aprovação a um repositório](#)
- [Gerenciar modelos de regras de aprovação](#)
- [Desassociar um modelo de regra de aprovação](#)
- [Excluir um modelo de regra de aprovação](#)

Criar um modelo de regra de aprovação

É possível criar um ou mais modelos de regras de aprovação para ajudá-lo a personalizar seus fluxos de trabalho de desenvolvimento em vários repositórios. Ao criar vários modelos, é possível configurar a criação automática de regras de aprovação para que diferentes ramificações tenham níveis apropriados de aprovações e controle. Por exemplo, é possível criar modelos diferentes para ramificações de produção e desenvolvimento e aplicar esses modelos a um ou mais repositórios. Quando os usuários criam solicitações pull nesses repositórios, a solicitação é avaliada em relação a esses modelos. Se a solicitação corresponder às condições nos modelos aplicados, serão criadas regras de aprovação da solicitação pull.

Você pode usar o console ou AWS CLI criar modelos de regras de aprovação. Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Tópicos

- [Criar um modelo de regra de aprovação \(console\)](#)
- [Criar um modelo de regra de aprovação \(AWS CLI\)](#)

Criar um modelo de regra de aprovação (console)

Os modelos de regras de aprovação não são associados a nenhum repositório por padrão. É possível fazer uma associação entre um modelo e um ou mais repositórios ao criar o modelo ou adicionar as associações mais tarde.

Como criar um modelo de regra de aprovação (console)

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. Escolha Approval rule templates (Modelos de regras de aprovação) e escolha Create template (Criar modelo).
3. Em Approval rule template name (Nome do modelo de regra de aprovação), dê ao modelo um nome descritivo para que você saiba para que serve. Por exemplo, se você deseja exigir que uma pessoa de um conjunto de desenvolvedores seniores aprove uma solicitação pull para que possa ser mesclada, é possível nomear a regra **Require 1 approver from a senior developer**.
4. (Opcional) Em Description (Descrição), forneça uma descrição da finalidade deste modelo. Isso pode ajudar outras pessoas a decidir se esse modelo é apropriado para os repositórios.
5. Em Number of approvals needed (Número de aprovações necessárias), insira o número desejado. O padrão é um.
6. (Opcional) Se deseja exigir que as aprovações de uma solicitação pull sejam provenientes de um grupo específico de usuários, em Approval rule members (Membros da regra de aprovação), escolha Add (Adicionar). Em Approver type (Tipo de aprovador), escolha uma das seguintes opções:
 - Nome de usuário do IAM ou perfil assumido: essa opção preenche previamente o ID da conta da Amazon Web Services para a conta que você usou para fazer login e requer apenas um nome. Ela pode ser usada para usuários de acesso federados e usuários do IAM cujos nomes correspondam ao nome fornecido. Essa é uma opção muito eficiente que oferece uma grande flexibilidade. Por exemplo, se você escolher essa opção, estiver conectado com a conta 123456789012 da Amazon Web Services e especificar **Mary_Major**, todas as opções a seguir contarão como aprovações provenientes desse usuário:
 - Um usuário do IAM na conta (arn:aws:iam::123456789012:user/Mary_Major)
 - Um usuário federado identificado no IAM como Mary_Major (arn:aws:sts::123456789012:federated-user/Mary_Major)

Essa opção não reconhece uma sessão ativa de alguém que assume a função de **CodeCommitReview** com um nome de sessão de função de Mary_Major (arn:aws:sts::123456789012:assumed-role/CodeCommitReview/Mary_Major), a menos que você inclua um curinga (*Mary_Major). Você também pode especificar o nome da função explicitamente (CodeCommitReview/Mary_Major).

- ARN totalmente qualificado: essa opção permite especificar o nome do recurso da Amazon (ARN) totalmente qualificado do perfil ou usuário do IAM. Essa opção também oferece suporte a funções assumidas usadas por outros AWS serviços, como AWS Lambda AWS CodeBuild e. Para funções assumidas, o formato do ARN deve ser

`arn:aws:sts::AccountID:assumed-role/RoleName` para funções (roles) e
`arn:aws:sts::AccountID:assumed-role/FunctionName` funções (functions).

Se você escolheu o nome de usuário do IAM ou perfil assumido como o tipo de aprovador, em Valor, insira o nome do usuário ou perfil do IAM ou o ARN totalmente qualificado do usuário ou perfil. Escolha Add (Adicionar) novamente para adicionar mais usuários ou funções, até que você tenha adicionado todos os usuários ou funções cujas aprovações contam para o número de aprovações necessárias.

Ambos os tipos de aprovadores permitem usar curingas (*) em seus valores. Por exemplo, se você escolher a opção nome de usuário do IAM ou perfil assumido e especificar **CodeCommitReview/***, todos os usuários que assumirem o perfil de **CodeCommitReview** serão contados no grupo de aprovação. Os nomes de sessão de função individual são contabilizados no número necessário de aprovadores. Dessa forma, tanto Mary_Major quanto Li_Juan contam como aprovações quando conectados e assumindo a função de CodeCommitReview. Para obter mais informações sobre IAM ARNs, curingas e formatos, consulte [Identificadores do IAM](#).

 Note

As regras de aprovação não oferecem suporte a aprovações entre contas.

7. (Opcional) Em Branch filters (Filtros de ramificação), insira nomes de ramificação de destino a serem usados para filtrar a criação de regras de aprovação. Por exemplo, se você especificar *main*, uma regra de aprovação será criada para pull requests em repositórios associados somente se a ramificação de destino da pull request for uma ramificação chamada *main*. É possível usar curingas (*) em nomes de ramificação para aplicar regras de aprovação a todos os nomes de ramificação que correspondam aos caracteres curinga. No entanto, não é possível usar um curinga no início de um nome de ramificação. É possível especificar até 100 nomes de ramificação. Se você não especificar nenhum filtro, o modelo se aplicará a todas as ramificações em um repositório associado.
8. (Opcional) Em Repositórios associados, na lista Repositórios, escolha os repositórios nessa Região da AWS que deseja associar a essa regra de aprovação.

 Note

É possível optar por associar repositórios após a criação do modelo. Para obter mais informações, consulte [Associar um modelo de regra de aprovação a um repositório](#).

9. Escolha Criar.

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 ✕

Criar um modelo de regra de aprovação (AWS CLI)

Você pode usar o AWS CLI para criar modelos de regras de aprovação. Ao usar o AWS CLI, você pode especificar referências de destino para o modelo, de forma que ele se aplique somente às pull requests cujas ramificações de destino correspondam às do modelo.

Como criar um modelo de regra de aprovação (AWS CLI)

1. Em um terminal ou uma linha de comando, execute o comando `create-approval-rule-template`, especificando:
 - O nome do modelo de regra de aprovação. Considere usar um nome que descreva sua finalidade.
 - Uma descrição do modelo de regra de aprovação. Assim como acontece com o nome, considere fornecer uma descrição detalhada.
 - A estrutura JSON do modelo de regra de aprovação. Essa estrutura pode incluir requisitos para referências de destino, que são ramificações de destino para solicitações pull às quais a regra de aprovação é aplicada e membros do grupo de aprovações, que são usuários cujas aprovações contam para o número de aprovações necessárias.

Ao criar o conteúdo da regra de aprovação, é possível especificar aprovadores em um grupo de aprovações de duas maneiras:

- **CodeCommitApprovers**: Essa opção requer somente uma conta e um recurso da Amazon Web Services. Ela pode ser usada para usuários de acesso federados e usuários do IAM cujo nome corresponde ao nome do recurso fornecido. Essa é uma opção muito eficiente que oferece uma grande flexibilidade. Por exemplo, se você especificar a conta 123456789012 da AWS e **Mary_Major**, todas as opções a seguir contam como aprovações provenientes desse usuário:
 - Um usuário do IAM na conta (`arn:aws:iam::123456789012:user/Mary_Major`)
 - Um usuário federado identificado no IAM como **Mary_Major** (`arn:aws:sts::123456789012:federated-user/Mary_Major`)

Essa opção não reconhece uma sessão ativa de alguém assumindo a função de *SeniorDevelopers* com um nome de sessão de função *Mary_Major* (`arn:aws:sts::123456789012:assumed-role/SeniorDevelopers/Mary_Major`), a menos que você inclua um caractere curinga (`*Mary_Major`).

- ARN totalmente qualificado: essa opção permite especificar o nome do recurso da Amazon (ARN) totalmente qualificado do perfil ou usuário do IAM.

Para obter mais informações sobre IAM ARNs, curingas e formatos, consulte [Identificadores do IAM](#).

O exemplo a seguir cria um modelo de regra de aprovação chamado **2-approver-rule-for-main** e uma descrição **Requires two developers from the team to approve the pull request if the destination branch is main**. O modelo exige que dois usuários que assumem a função de **CodeCommitReview** aprovem qualquer solicitação pull para que ela possa ser mesclada à ramificação **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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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"
  }
}
```


Associar um modelo de regra de aprovação a um repositório

Os modelos de regras de aprovação são criados em um local específico Região da AWS, mas não afetam nenhum repositório Região da AWS até serem associados. Para aplicar um modelo a um ou mais repositórios, é necessário associar o modelo ao repositório ou repositórios. É possível aplicar um único modelo a vários repositórios em uma Região da AWS. Isso ajuda a automatizar e padronizar o fluxo de trabalho de desenvolvimento em seus repositórios, criando condições consistentes para aprovar e mesclar solicitações pull.

Só é possível associar um modelo de regra de aprovação a repositórios na Região da AWS onde o modelo de regra de aprovação foi criado.

Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Tópicos

- [Associar um modelo de regra de aprovação \(console\)](#)
- [Associar um modelo de regra de aprovação \(AWS CLI\)](#)

Associar um modelo de regra de aprovação (console)

Talvez você tenha repositórios associados a um modelo de regra de aprovação ao criá-lo. (Esta etapa é opcional.) É possível adicionar ou remover associações editando o modelo.

Como associar um modelo de regra de aprovação a repositórios

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Escolha Approval rule templates (Modelos de regras de aprovação). Escolha o modelo e escolha Edit (Editar).
3. Em Associated Repositories (Repositórios associados), escolha os repositórios na lista Repositories (Repositórios). Cada repositório associado é exibido na caixa de listagem.
4. Escolha Salvar. As regras de aprovação agora são aplicadas a solicitações pull criadas nesses repositórios associados.

Associar um modelo de regra de aprovação (AWS CLI)

Você pode usar o AWS CLI para associar um modelo de regra de aprovação a um ou mais repositórios.

Como associar um modelo a um único repositório

1. No terminal ou na linha de comando, execute o comando `associate-approval-rule-template-with-repository`, especificando:
 - O nome do modelo de regra de aprovação que você deseja associar a um repositório.
 - O nome do repositório a ser associado ao modelo de regra de aprovação.

Por exemplo, para associar um modelo de regra de aprovação nomeado *2-approver-rule-for-main* a um repositório chamado *MyDemoRepo*:

```
aws codecommit associate-approval-rule-template-with-repository --repository-name MyDemoRepo --approval-rule-template-name 2-approver-rule-for-main
```

2. Se houver êxito, o comando não retornará nada.

Como associar um modelo a vários repositórios

1. No terminal ou na linha de comando, execute o comando `batch-associate-approval-rule-template-with-repositories`, especificando:
 - O nome do modelo de regra de aprovação que você deseja associar a um repositório.
 - Os nomes dos repositórios a serem associados ao modelo de regra de aprovação.

Por exemplo, para associar um modelo de regra de aprovação chamado **2-approver-rule-for-main** a um repositório chamado **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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "associatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Gerenciar modelos de regras de aprovação

Você pode gerenciar os modelos de regras de aprovação em um Região da AWS para ajudar a entender como eles estão sendo usados e para que servem. Por exemplo, é possível editar os nomes e as descrições dos modelos de regras de aprovação para ajudar outras pessoas a entenderem sua finalidade. Você pode listar todos os modelos de regras de aprovação em um Região da AWS e obter informações sobre o conteúdo e a estrutura de um modelo. É possível revisar quais modelos estão associados a um repositório e quais repositórios estão associados a um modelo.

Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Gerenciar modelos de regras de aprovação (console)

Você pode visualizar e gerenciar seus modelos de regras de aprovação no CodeCommit console.

Como gerenciar modelos de regras de aprovação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Escolha Modelos de regras de aprovação para ver a lista de modelos de regras de aprovação no Região da AWS local em que você está conectado.

Note

Os modelos de regras de aprovação só estão disponíveis no Região da AWS local em que foram criados.

3. Se você deseja fazer alterações em um modelo, escolha-o na lista e escolha Edit (Editar).

4. Faça suas alterações e escolha Salvar.

Gerenciar modelos de regras de aprovação (AWS CLI)

É possível gerenciar seus modelos de regra de aprovação com os seguintes comandos da AWS CLI :

- [list-approval-rule-templates](#), para ver uma lista de todos os modelos de regras de aprovação em um Região da AWS
- [get-approval-rule-template](#), para visualizar o conteúdo de um modelo de regra de aprovação
- [update-approval-rule-template-content](#), para alterar o conteúdo de um modelo de regra de aprovação
- [update-approval-rule-template-name](#), para alterar o nome de um modelo de regra de aprovação
- [update-approval-rule-template-description](#), para alterar a descrição de um modelo de regra de aprovação
- [list-repositories-for-approval-rule-template](#), para visualizar todos os repositórios associados a um modelo de regra de aprovação
- [list-associated-approval-rule-templates-for-repository](#), para visualizar todos os modelos de regras de aprovação associados a um repositório

Para listar todos os modelos de regras de aprovação em um Região da AWS

1. No terminal ou na linha de comando, execute o comando `list-approval-rule-templates`. Por exemplo, para listar todos os modelos de regras de aprovação na região Leste dos EUA (Ohio):

```
aws codecommit list-approval-rule-templates --region us-east-2
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Para obter o conteúdo de um modelo de regra de aprovação

1. No terminal ou na linha de comando, execute o comando `get-approval-rule-template`, especificando o nome do modelo de regra de aprovação:

```
aws codecommit get-approval-rule-template --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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.\"
  }
}
```

Para atualizar o conteúdo de um modelo de regra de aprovação

1. No terminal ou prompt de comando, execute o comando `update-approval-rule-template-content`, especificando o nome do modelo e o conteúdo alterado. Por exemplo, para alterar o conteúdo de um modelo de regra de aprovação chamado **1-approver-rule** para redefinir o grupo de aprovações para usuários que assumem a função de **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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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"
  }
}
```

Para atualizar o nome de um modelo de regra de aprovação

1. No terminal ou prompt de comando, execute o comando `update-approval-rule-template-name`, especificando o nome atual e o nome para o qual você deseja alterá-lo. Por exemplo, para alterar o nome de um modelo de regra de aprovação de **1-approver-rule** para **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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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"
    }
}

```

Para atualizar a descrição de um modelo de regra de aprovação

1. No terminal ou na linha de comando, execute o comando `update-approval-rule-template-description`, especificando o nome do modelo de regra de aprovação e a nova descrição:

```

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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```

{
  "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"
  }
}

```

Para listar todos os repositórios associados a um modelo

1. Na linha de comando ou terminal, execute o comando `list-repositories-for-approval-rule-template`, especificando o nome do modelo:

```

aws codecommit list-repositories-for-approval-rule-template --approval-rule-
template-name 2-approver-rule-for-main

```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

Para listar todos os modelos associados a um repositório

1. Na linha de comando ou terminal, execute o comando `list-associated-approval-rule-templates-for-repository`, especificando o nome do repositório:

```
aws codecommit list-associated-approval-rule-templates-for-repository --repository-name MyDemoRepo
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Desassociar um modelo de regra de aprovação

Se as regras de aprovação geradas por um modelo de regra de aprovação não fizerem mais sentido para o fluxo de trabalho de sua equipe em um repositório, será possível desassociar o modelo desse repositório. A desassociação de um modelo não remove regras de aprovação criadas enquanto o modelo esteve associado ao repositório.

Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Desassociar um modelo de regra de aprovação (console)

É possível usar o console para remover a associação entre um repositório e um modelo de regra de aprovação.

Como desassociar um modelo de regra de aprovação de repositórios

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Escolha Approval rule templates (Modelos de regras de aprovação). Escolha o modelo que deseja desassociar de um repositório ou repositórios e escolha Edit (Editar).
3. Em Associated repositories (Repositórios associados), escolha o X ao lado dos repositórios que deseja desassociar. Os nomes do repositório não são mais exibidos.
4. Escolha Salvar. As regras de aprovação não são aplicadas a solicitações pull criadas nesses repositórios. As regras ainda são aplicadas a solicitações pull que foram criadas enquanto a associação estava em vigor.

Desassociar um modelo de regra de aprovação (AWS CLI)

Você pode usar o AWS CLI para desassociar um ou mais repositórios de um modelo de regra de aprovação.

Como desassociar um modelo de regra de aprovação de um repositório

1. No terminal ou na linha de comando, execute o comando `disassociate-approval-rule-template-from-repository`, especificando:
 - O nome do modelo de regra de aprovação.
 - O nome do repositório.

Por exemplo, para desassociar um modelo de regra de aprovação chamado **1-approver-rule-for-all-pull-requests** de um repositório chamado **MyDemoRepo**:

```
aws codecommit disassociate-approval-rule-template-from-repository --repository-name MyDemoRepo --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

2. Se houver êxito, o comando não retornará nada.

Como desassociar um modelo de regra de aprovação de vários repositórios

1. No terminal ou na linha de comando, execute o comando `batch-disassociate-approval-rule-template-from-repositories`, especificando:
 - O nome do modelo de regra de aprovação.
 - Os nomes dos repositórios.

Por exemplo, para um modelo de regra de aprovação chamado **1-approver-rule-for-all-pull-requests** de um repositório chamado **MyDemoRepo** e um repositório chamado **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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "disassociatedRepositoryNames": [  
    "MyDemoRepo",  
    "MyOtherDemoRepo"  
  ],  
  "errors": []  
}
```

Excluir um modelo de regra de aprovação

É possível excluir um modelo de regra de aprovação se não o estiver usando em nenhum repositório. A exclusão de modelos de regras de aprovação não usados ajuda a manter seus modelos organizados e facilita a localização de modelos que façam sentido para seus fluxos de trabalho.

Para obter mais informações sobre políticas e permissões gerenciadas para modelos de regras de aprovação, consulte [Permissões para ações em modelos de regras de aprovação](#) e [AWS políticas gerenciadas para CodeCommit](#).

Tópicos

- [Excluir um modelo de regra de aprovação \(console\)](#)

- [Excluir um modelo de regra de aprovação \(AWS CLI\)](#)

Excluir um modelo de regra de aprovação (console)

É possível excluir um modelo de regra de aprovação se não for mais relevante para seu trabalho de desenvolvimento. Ao usar o console para excluir um modelo de regra de aprovação, ele é desassociado de qualquer repositório durante o processo de exclusão.

Para excluir um modelo de regra de aprovação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Escolha Approval rule templates (Modelos de regras de aprovação). Escolha o modelo que você deseja excluir e escolha Delete (Excluir).

Excluir um modelo de regra de aprovação (AWS CLI)

Você pode usar o AWS CLI para excluir uma regra de aprovação se ela tiver sido desassociada de todos os repositórios. Para obter mais informações, consulte [Desassociar um modelo de regra de aprovação \(AWS CLI\)](#).

Para excluir um modelo de regra de aprovação

1. Em um terminal ou uma linha de comando, execute o comando `delete-approval-rule-template`, especificando o nome do modelo de regra de aprovação que deseja excluir:

```
aws codecommit delete-approval-rule-template --approval-rule-template-name 1-approver-for-all-pull-requests
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte. Se o modelo de regra de aprovação já foi excluído, esse comando não retornará nada.

```
{
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

Trabalhando com confirmações em repositórios AWS CodeCommit

As confirmações são snapshots dos conteúdos e das alterações feitas nos conteúdos do repositório. Sempre que um usuário confirmar e enviar uma alteração por push, essa informação será salva e armazenada. E também as informações que incluem quem confirmou a alteração, a data e a hora da confirmação e as alterações feitas como parte da confirmação. Você também pode adicionar tags a confirmações para identificar facilmente confirmações específicas. Em CodeCommit, você pode:

- Analisar confirmações.
- Visualizar o histórico de confirmações em um gráfico.
- Comparar uma confirmação com seu pai ou outro especificador.
- Adicionar comentários a suas confirmações e responder aos comentários feitos por outros.



Antes de enviar commits para um CodeCommit repositório, você deve configurar seu computador local para se conectar ao repositório. Para o método mais simples, consulte [Para usuários de HTTPS usando credenciais do Git](#).

Para obter informações sobre como trabalhar com outros aspectos do seu repositório em CodeCommit [Trabalhar com repositórios](#), consulte [Trabalhar com arquivos](#), [Trabalhar com solicitações pull](#), [Como trabalhar com ramificações](#), e [Trabalhar com preferências do usuário](#)

Tópicos

- [Crie um commit em AWS CodeCommit](#)
- [Exibir detalhes do commit em AWS CodeCommit](#)
- [Compare os commits em AWS CodeCommit](#)
- [Comente em um commit em AWS CodeCommit](#)
- [Crie uma tag Git em AWS CodeCommit](#)
- [Veja os detalhes da tag Git em AWS CodeCommit](#)
- [Excluir uma tag Git em AWS CodeCommit](#)

Crie um commit em AWS CodeCommit

Ao criar o primeiro commit para um novo repositório, você usa o put-file comando AWS CLI and the. Isso cria a primeira confirmação e permite que você crie e especifique a ramificação padrão para seu novo repositório. Você pode usar o Git ou o AWS CLI para criar um commit em um CodeCommit repositório. Se o repositório local estiver conectado a um CodeCommit repositório, você usa o Git para enviar o commit do repositório local para o repositório. CodeCommit Para criar um commit diretamente no CodeCommit console, consulte [Criar ou adicionar um arquivo para um repositório do AWS CodeCommit](#) [Editar o conteúdo de um arquivo em um AWS CodeCommit repositório](#) e.

Note

Como prática recomendada, recomendamos que você use as versões mais recentes suportadas do AWS CLI, do Git e de outros softwares. Se você usar o AWS CLI, verifique se você tem uma versão recente instalada para garantir que está usando uma versão que contém o `create-commit` comando.

Tópicos

- [Crie o primeiro commit para um repositório usando o AWS CLI](#)
- [Criar uma confirmação usando um cliente Git](#)
- [Crie um commit usando o AWS CLI](#)

Crie o primeiro commit para um repositório usando o AWS CLI

Você pode usar o `put-file` comando AWS CLI and the para criar seu primeiro commit para um repositório. Usar `put-file` cria uma primeira confirmação que adiciona um arquivo ao repositório vazio e cria uma ramificação com o nome que você especificar. Ele designa a nova ramificação como a ramificação padrão do seu repositório.

Note

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para criar o primeiro commit para um repositório usando o AWS CLI

1. No seu computador local, crie o arquivo que você deseja adicionar como o primeiro arquivo ao CodeCommit repositório. Uma prática comum é criar um arquivo markdown README .md que explique a finalidade desse repositório para outros usuários do repositório. Se você incluir um README .md arquivo, o conteúdo do arquivo será exibido automaticamente na parte inferior da página de código do seu repositório no CodeCommit console.
2. No terminal ou na linha de comando, execute o comando `put-file`, especificando:
 - O nome do repositório no qual você deseja adicionar o primeiro arquivo.
 - O nome da ramificação que você deseja criar como a ramificação padrão.
 - A localização atual do arquivo. A sintaxe usada para essa localização varia, dependendo do seu sistema operacional local.
 - O nome do arquivo que você deseja adicionar, incluindo o caminho onde o arquivo atualizado está armazenado no repositório.
 - O nome de usuário e e-mail a serem associados a este arquivo.
 - Uma mensagem de confirmação explicando por que você adicionou este arquivo.

O nome de usuário, endereço de e-mail e a mensagem de confirmação são opcionais, mas podem ajudar outros usuários a saber quem fez a alteração e por quê. Se você não fornecer um nome de usuário, o CodeCommit padrão é usar seu nome de usuário do IAM ou uma derivação do login do console como nome do autor.

Por exemplo, para adicionar um arquivo chamado *README.md* com um exemplo de conteúdo de arquivo codificado na base 6 a um repositório com o nome de uma ramificação chamada *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "commitId": "724caa36EXAMPLE",
  "blobId": "a8a94062EXAMPLE",
  "treeId": "08b2fc73EXAMPLE"
}
```

Criar uma confirmação usando um cliente Git

Você pode criar commits usando um cliente Git instalado no seu computador local e, em seguida, enviar esses commits para o seu repositório. CodeCommit

1. Conclua os pré-requisitos, incluindo [Configurando](#).

Important

Se você não concluiu a configuração, não poderá se conectar ou enviar confirmações ao repositório usando o Git.

2. Verifique se você está criando uma confirmação na ramificação correta. Para ver uma lista de ramificações disponíveis e descobrir qual delas está configurada para seu uso no momento, execute `git branch`. Todas as ramificações são exibidas. Um asterisco (*) aparece ao lado de sua ramificação atual. Para trocar para uma ramificação diferente, execute `git checkout branch-name`. Se for sua primeira confirmação, execute o comando `git config` para configurar seu cliente Git para criar uma ramificação inicial com o nome que você deseja usar para essa ramificação. Por exemplo, se você quiser que sua ramificação padrão tenha o nome *development*:

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

 Tip

Esse comando só está disponível na Git versão 2.28 e posterior.
Você também pode executar esse comando para definir o nome da ramificação padrão como **development** para todos os repositórios recém-criados:

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

3. Faça uma alteração na ramificação (como adicionar, modificar ou excluir um arquivo).

Por exemplo, no repositório local, crie um arquivo chamado `bird.txt` com o seguinte texto:

```
bird.txt
-----
Birds (class Aves or clade Avialae) are feathered, winged, two-legged, warm-
blooded, egg-laying vertebrates.
```

4. Execute `git status`, que indicará que `bird.txt` ainda não foi incluído em nenhuma confirmação pendente:

```
...
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    bird.txt
```

5. Execute `git add bird.txt` para incluir o novo arquivo na confirmação pendente.
6. Se você executar `git status` novamente, visualizará uma saída semelhante à seguinte. Isso indica que `bird.txt` agora faz parte da confirmação pendente ou está preparado para a confirmação:

```
...
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   bird.txt
```


7. Para finalizar a confirmação, execute `git commit` com a opção `-m` (por exemplo, `git commit -m "Adding bird.txt to the repository."`) A opção `-m` cria a mensagem de confirmação.
8. Se você executar `git status` novamente, visualizará uma saída semelhante à seguinte. Isso indica que o commit está pronto para ser enviado do repositório local para o CodeCommit repositório:

```
...
nothing to commit, working directory clean
```

9. Antes de enviar a confirmação finalizada do repositório local para o CodeCommit repositório, você pode ver o que está enviando executando `git diff --stat remote-name/branch-name`, onde `remote-name` está o apelido que o repositório local usa para o CodeCommit repositório e o nome da ramificação a ser `branch-name` comparada.

Tip

Para obter o apelido, execute `git remote`. Para obter uma lista de nomes de ramificação, execute `git branch`. Um asterisco (*) aparece ao lado da ramificação atual. Você também pode executar `git status` para obter o nome da ramificação atual.

Note

Se você clonou o repositório, do ponto de vista do repositório local, não `remote-name` é o nome do repositório. CodeCommit Quando você clona um repositório, `remote-name` é definido automaticamente como `origin`

Por exemplo, `git diff --stat origin/main` exibiria uma saída semelhante à seguinte:

```
bird.txt | 1 +
1 file changed, 1 insertion(+)
```

A saída pressupõe que você já tenha conectado o repositório local ao CodeCommit repositório. (Para obter instruções, consulte [Conexão a um repositório](#).)

10. Quando você estiver pronto para enviar o commit do repositório local para o CodeCommit repositório, execute `git push remote-name branch-name`, onde `remote-name` está o apelido

que o repositório local usa para o CodeCommit repositório e o nome da ramificação **branch-name** a ser enviada para o repositório. CodeCommit

Por exemplo, executar `git push origin main` exibiria uma saída semelhante à seguinte:

Para HTTPS:

```
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 516 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
    b9e7aa6..3dbf4dd main -> main
```

Para o 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 você adicionar a opção `-u` para `git push` (por exemplo, `git push -u origin main`), você só precisará executar `git push` no futuro, porque as informações de rastreamento de upstream foram definidas. Para obter as informações de rastreamento upstream, execute `git remote show remote-name` (por exemplo, `git remote show origin`).

Para obter mais opções, consulte a documentação do Git.

Crie um commit usando o AWS CLI

Você pode usar o `create-commit` comando AWS CLI and the para criar um commit para um repositório na ponta de uma ramificação especificada. Também é possível criar uma confirmação de mesclagem sem referência para representar os resultados da mesclagem de dois especificadores de confirmação. Para obter mais informações, consulte [Criar uma confirmação sem referência](#).

Note

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para criar uma confirmação

1. No computador local, faça as alterações que você deseja que sejam confirmadas no repositório do CodeCommit.
2. No terminal ou na linha de comando, execute o comando `create-commit`, especificando:
 - O repositório no qual você deseja confirmar as alterações.
 - A ramificação na qual você deseja confirmar as alterações.
 - O ID de confirmação completo da confirmação mais recente feita nessa ramificação, também conhecido como a confirmação de ponta ou cabeça ou o ID de confirmação pai.
 - Se deseja manter pastas vazias, caso as alterações feitas excluam o conteúdo dessas pastas. Por padrão, esse valor é `false` (falso).
 - As informações sobre os arquivos que você deseja adicionar, alterar ou excluir.
 - O nome de usuário e o e-mail a serem associados a essas alterações.
 - Uma mensagem de confirmação que explica por que as alterações foram feitas.

O nome de usuário, o endereço de e-mail e a mensagem de confirmação são opcionais, mas ajudam outros usuários a saber quem fez as alterações e por que. Se você não fornecer um nome de usuário, o CodeCommit padrão é usar seu nome de usuário do IAM ou uma derivação do login do console como nome do autor.

Por exemplo, para criar um commit para um repositório que adiciona um README.md arquivo a um repositório nomeado *MyDemoRepo* na *main* ramificação. O conteúdo do arquivo está em Base64 e diz “Boas-vindas ao repositório da nossa equipe!”:

```
aws codecommit create-commit --repository-name MyDemoRepo --  
branch-name main --parent-commit-id 4c925148EXAMPLE --put-files  
"filePath=README.md,fileContent=V2VsY29tZSB0byBvdXIgdGVhbSBvZXBvc2l0b3J5IQo="
```

 Tip

Para obter o ID da confirmação principal, execute o comando [get-branch](#).

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "commitId": "4df8b524-EXAMPLE",  
  "treeId": "55b57003-EXAMPLE",  
  "filesAdded": [  
    {  
      "blobId": "5e1c309dEXAMPLE",  
      "absolutePath": "meeting.md",  
      "fileMode": "NORMAL"  
    }  
  ],  
  "filesDeleted": [],  
  "filesUpdated": []  
}
```

Para criar um commit que faz alterações em arquivos chamados *file1.py* e *file2.txt*, renomeia um arquivo de *picture.png* para *image1.png* e o move de um diretório nomeado *pictures* para um diretório chamado *images*, e exclui um arquivo nomeado *ExampleSolution.py* em um repositório chamado *MyDemoRepo* em uma ramificação chamada *MyFeatureBranch* cujo commit mais recente tem uma ID de: *4c925148EXAMPLE*

```
aws codecommit create-commit --repository-name MyDemoRepo --branch-  
name MyFeatureBranch --parent-commit-id 4c925148EXAMPLE --author-name "Saanvi  
Sarkar"
```

```
--email "saanvi_sarkar@example.com" --commit-message "I'm creating this commit to
update a variable name in a number of files."
--keep-empty-folders false --put-files '{"filePath": "file1.py", "fileMode":
"EXECUTABLE", "fileContent": "bucket_name = sys.argv[1] region = sys.argv[2]}"'
'{"filePath": "file2.txt", "fileMode": "NORMAL", "fileContent": "//Adding a comment
to explain the variable changes in file1.py"}' '{"filePath": "images/image1.png",
"fileMode": "NORMAL", "sourceFile": {"filePath": "pictures/picture.png", "isMove":
true}}' --delete-files filePath="ExampleSolution.py"
```

Note

A sintaxe para o segmento `--put-files` varia dependendo do seu sistema operacional. O exemplo acima é otimizado para usuários Linux, macOS, Unix e usuários do Windows com um emulador de Bash. Os usuários do Windows na linha de comando ou no Powershell devem usar uma sintaxe adequada para esses sistemas.

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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"  
  }  
]  
}
```

Exibir detalhes do commit em AWS CodeCommit

Você pode usar o AWS CodeCommit console para pesquisar o histórico de confirmações em um repositório. Isso pode ajudá-lo a identificar as alterações feitas em um repositório, incluindo:

- Quando e por quem as alterações foram feitas.
- Quando confirmações específicas foram mescladas em determinada ramificação.

Visualizar o histórico de confirmações de uma ramificação também pode ajudar a entender a diferença entre ramificações. Se você usa tags, pode visualizar rapidamente a confirmação marcada com uma tag e os pais dessa confirmação marcada. Na linha de comando, você pode usar o Git para ver detalhes sobre os commits em um repositório ou repositório local. CodeCommit

Navegar pelas confirmações em um repositório

Você pode usar o AWS CodeCommit console para pesquisar o histórico de confirmações em um repositório. Você também pode visualizar o gráfico de confirmações no repositório e suas ramificações ao longo do tempo. Isso pode ajudá-lo a entender o histórico do repositório, incluindo quando as alterações foram feitas.

 Note

Usar o comando git rebase para rebasear um repositório altera o histórico dele, o que pode fazer com que as confirmações apareçam fora de ordem. Para obter mais informações, consulte [Git Branching - Rebasing](#) na documentação do Git.

Tópicos

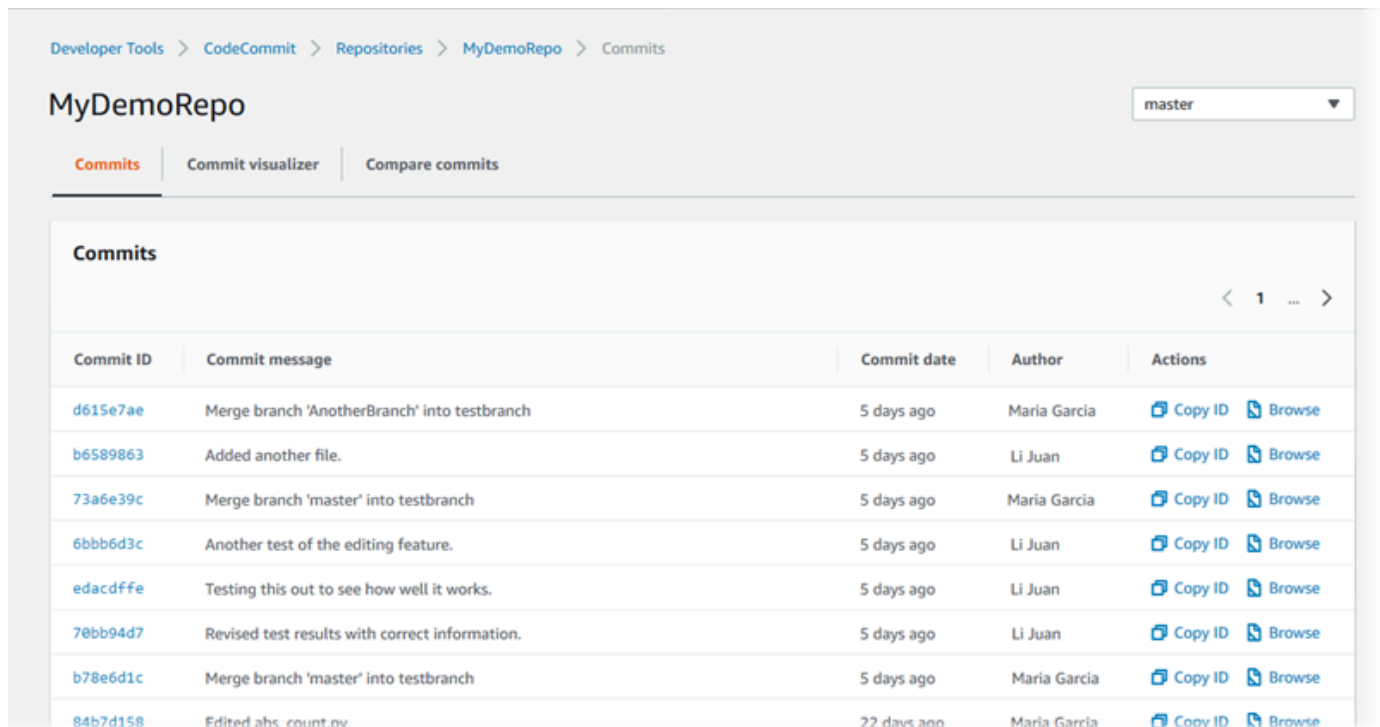
- [Navegar pelo histórico de confirmações de um repositório](#)
- [Visualizar um gráfico do histórico de confirmações de um repositório](#)

Navegar pelo histórico de confirmações de um repositório

É possível navegar pelo histórico de confirmações de uma ramificação ou tag específica do repositório, incluindo informações sobre quem confirmou e a mensagem de confirmação. Você também pode visualizar o código de uma confirmação.

Para navegar pelo histórico de confirmações

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório do qual você deseja rever o histórico de confirmação.
3. No painel de navegação, selecione Commits. Na exibição do histórico de confirmações, é exibido um histórico de confirmações do repositório na ramificação padrão em ordem cronológica reversa da data da confirmação. A data e a hora estão em Tempo Universal Coordenado (UTC). É possível visualizar o histórico de confirmações de uma ramificação diferente escolhendo o botão seletor de visualização e, em seguida, selecionando uma ramificação na lista. Se você estiver usando tags no repositório, será possível visualizar uma confirmação com uma tag específica e seus pais ao selecioná-la no botão seletor de visualização.



4. Para exibir a diferença entre uma confirmação e seu pai e para ver qualquer comentário sobre as alterações, escolha o ID de confirmação abreviado. Para obter mais informações, consulte [Comparar uma confirmação a seu pai](#) e [Comentar sobre uma confirmação](#). Para visualizar a diferença entre uma confirmação e qualquer outro especificador de confirmação, incluindo ramificação, tag ou ID de confirmação, consulte [Comparação entre dois especificadores de confirmação](#).
5. Siga um dos procedimentos abaixo:
 - Para visualizar a data e hora em que uma alteração foi feita, passe o mouse sobre a data da confirmação.
 - Para visualizar o ID completo da confirmação, copie-o e cole-o em um editor de texto ou outro local. Para copiá-lo, selecione Copy ID (Copiar ID).
 - Para visualizar o código como estava no momento de uma confirmação, escolha Browse (Procurar). Os conteúdos do repositório como foram no momento da confirmação são exibidos na exibição de código. O botão seletor de visualização exibe o ID abreviado da confirmação em vez de uma ramificação ou tag.

Visualizar um gráfico do histórico de confirmações de um repositório

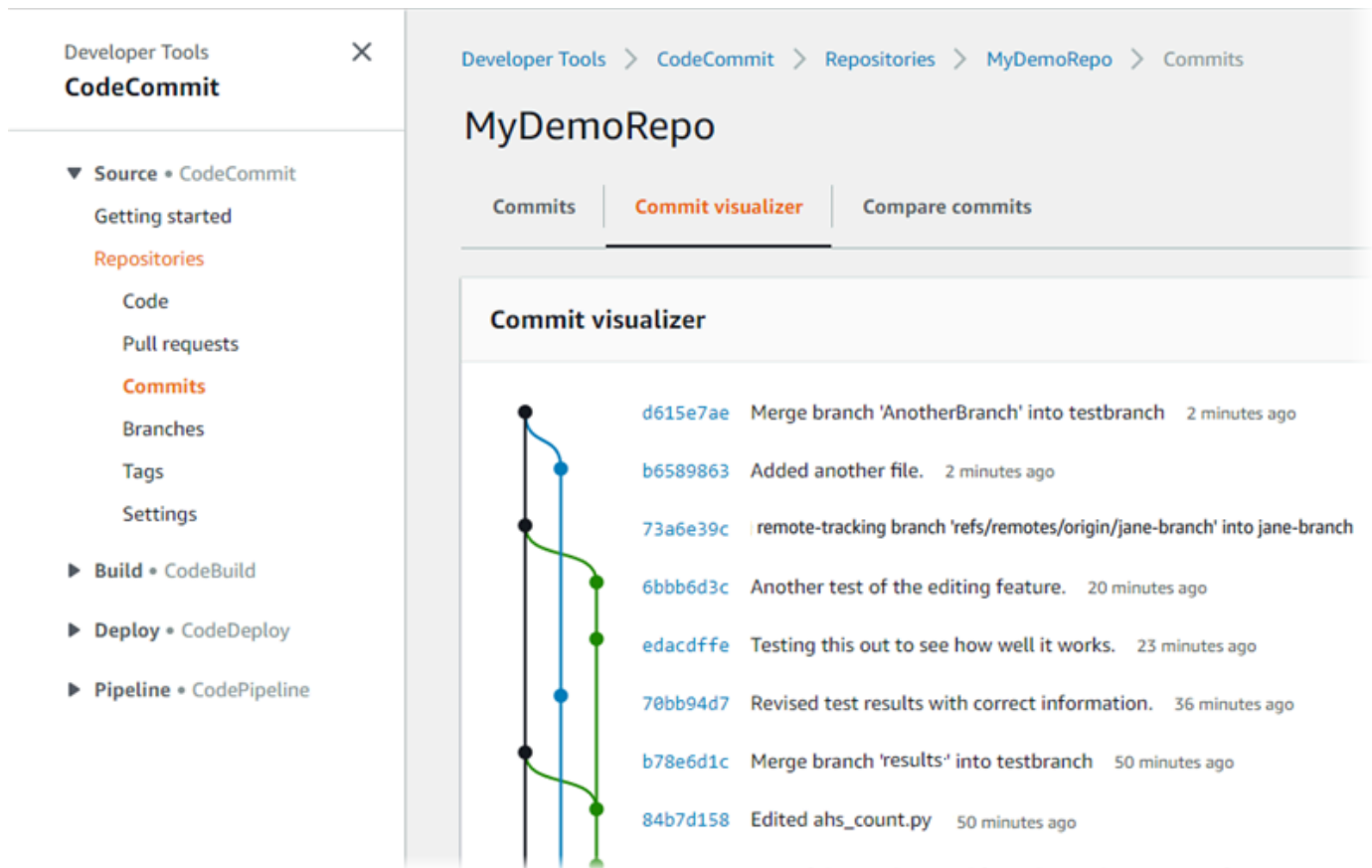
É possível visualizar um gráfico das confirmações feitas em um repositório. A exibição do visualizador de confirmações é a representação de um gráfico acíclico dirigido (DAG) de todas as confirmações feitas em uma ramificação do repositório. Essa representação gráfica pode ajudar a entender quando confirmações e recursos associados foram adicionados ou mesclados. Ela também pode ajudar a identificar quando uma alteração foi feita em relação a outras alterações.

Note

As confirmações mescladas usando o método `fastForward` não aparecem como linhas separadas no gráfico de confirmações.

Para visualizar um gráfico de confirmações

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório do qual você deseja visualizar um gráfico de confirmação.
3. No painel de navegação, selecione Commits (Confirmações) e, em seguida, escolha a guia Commit visualizer (Confirmar visualizador).



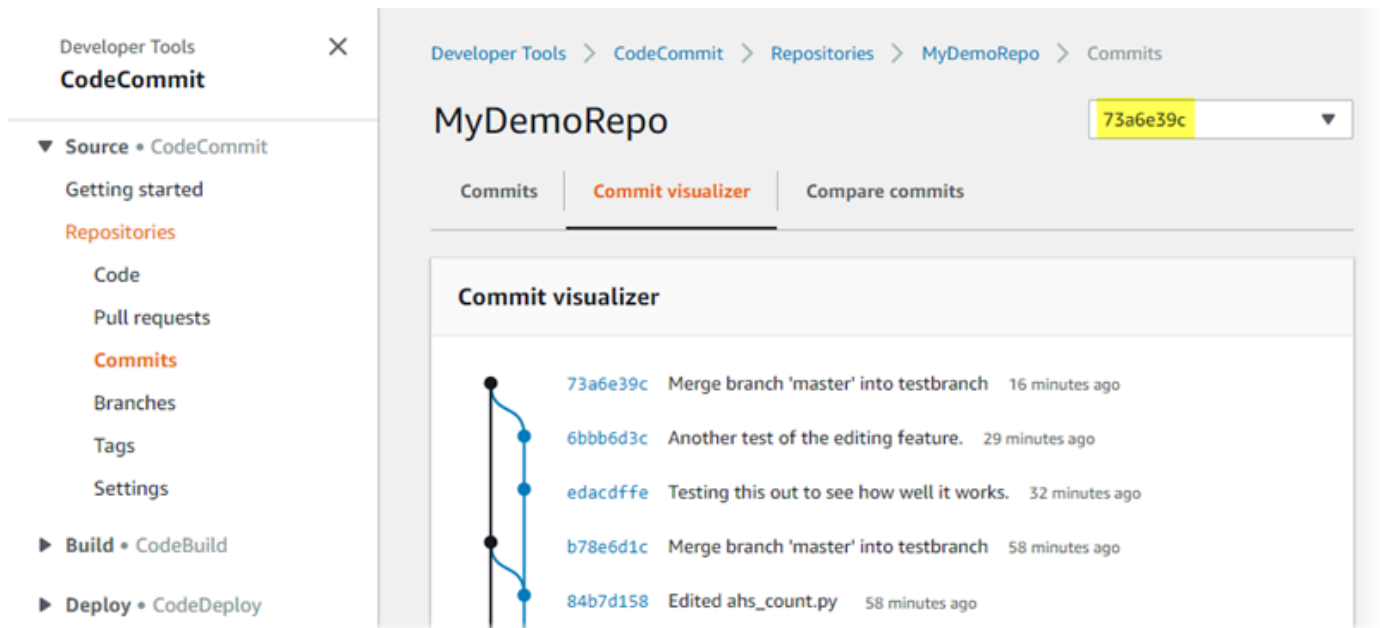
No gráfico de confirmações, o ID da confirmação abreviado e o assunto de cada mensagem de confirmação aparecem ao lado desse ponto no gráfico.

Note

O gráfico pode exibir até 35 ramificações em uma página. Se houver mais de 35 ramificações, o gráfico será muito complexo para ser exibido. Você pode simplificar a exibição de duas formas:

- Usando o botão seletor de visualização para mostrar o gráfico de uma ramificação específica.
- Colando um ID completo de confirmação na caixa de pesquisa para renderizar o gráfico da confirmação.

4. Para renderizar um novo gráfico de uma confirmação, escolha o ponto no gráfico que corresponde a essa confirmação. O botão seletor de visualização altera para o ID abreviado da confirmação.



Visualizar detalhes de confirmação (AWS CLI)

O Git permite visualizar detalhes sobre confirmações. Você também pode usar o AWS CLI para ver detalhes sobre os commits em um repositório local ou em um CodeCommit repositório executando os seguintes comandos:

- Para visualizar informações sobre uma confirmação, execute [aws codecommit get-commit](#).
- Para visualizar informações sobre várias confirmações, execute [aws codecommit batch-get-commits](#).
- Para visualizar informações sobre uma confirmação de mesclagem, execute [aws codecommit get-merge-commit](#).
- Para ver informações sobre alterações em um especificador de confirmação (ramificação, tag, HEAD ou outras referências totalmente qualificadas, como confirmação IDs), execute [aws codecommit get-differences](#).
- Para visualizar o conteúdo codificado em base64 de um objeto blob do Git em um repositório, execute [aws codecommit get-blob](#).

Como visualizar informações sobre uma confirmação

1. Execute o comando `aws codecommit get-commit`, especificando:

- O nome do CodeCommit repositório (com a `--repository-name` opção).
- O ID completo da confirmação.

Por exemplo, para visualizar informações sobre um commit com o ID 317f8570EXAMPLE em um CodeCommit repositório chamado MyDemoRepo:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id
317f8570EXAMPLE
```

2. Se houver êxito, a saída desse comando incluirá o seguinte:

- Informações sobre o autor da confirmação (como configurado no Git), incluindo a data no formato time stamp e o deslocamento do Tempo Universal Coordenado (UTC).
- Informações sobre o responsável pela confirmação (como configurado no Git), incluindo a data no formato time stamp e o deslocamento do UTC.
- O ID da árvore do Git onde existe a confirmação.
- O ID de confirmação da confirmação pai.
- A mensagem de confirmação.

Veja um exemplo de saída, com base no comando do exemplo anterior:

```
{
  "commit": {
    "additionalData": "",
    "committer": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "author": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "treeId": "347a3408EXAMPLE",
    "parents": [
      "4c925148EXAMPLE"
    ],
```

```
    "message": "Fix incorrect variable name"
  }
}
```

Como visualizar informações sobre uma confirmação de mesclagem

1. Execute o comando `get-merge-commit`, especificando:

- Um especificador de confirmação para a origem da mesclagem (com a opção `--source-commit-specifier`).
- Um especificador de confirmação para o destino da mesclagem (com a opção `--destination-commit-specifier`).
- A opção de mesclagem a ser usada (com a opção `--merge-option`).
- O nome do repositório (com a opção `--repository-name`).

Por exemplo, para visualizar informações sobre uma confirmação de mesclagem para a ramificação de origem nomeada *bugfix-bug1234* com uma ramificação de destino nomeada *main* usando a *THREE_WAY_MERGE* estratégia em um repositório chamado: *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 houver êxito, a saída desse comando retornará informações semelhantes a estas:

```
{
  "sourceCommitId": "c5709475EXAMPLE",
  "destinationCommitId": "317f8570EXAMPLE",
  "baseCommitId": "fb12a539EXAMPLE",
  "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

Como visualizar informações sobre várias confirmações

1. Execute o comando `batch-get-commits`, especificando:

- O nome do CodeCommit repositório (com a `--repository-name` opção).

- Uma lista de confirmações completas IDs para cada confirmação sobre a qual você deseja ver informações.

Por exemplo, para ver informações sobre confirmações com o IDs 317f8570EXAMPLE e 4c925148EXAMPLE em um CodeCommit repositório chamado: MyDemoRepo

```
aws codecommit batch-get-commits --repository-name MyDemoRepo --commit-ids
317f8570EXAMPLE 4c925148EXAMPLE
```

2. Se houver êxito, a saída desse comando incluirá o seguinte:

- Informações sobre os autores das confirmações (como configurado no Git), incluindo a data no formato time stamp e o deslocamento do Tempo Universal Coordenado (UTC).
- Informações sobre os responsáveis pelas confirmações (como configurado no Git), incluindo a data no formato time stamp e o deslocamento do UTC.
- A IDs da árvore Git em que o commit existe.
- O compromisso IDs do pai.
- As mensagens de confirmação.

Veja um exemplo de saída, com base no comando do exemplo anterior:

```
{
  "commits": [
    {
      "additionalData": "",
      "committer": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "author": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "commitId": "317f8570EXAMPLE",
      "treeId": "1f330709EXAMPLE",
      "parents": [
        "6e147360EXAMPLE"
      ]
    }
  ]
}
```

```
    ],
    "message": "Change variable name and add new response element"
  },
  {
    "additionalData": "",
    "committer": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "author": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "commitId": "4c925148EXAMPLE",
    "treeId": "1f330709EXAMPLE",
    "parents": [
      "317f8570EXAMPLE"
    ],
    "message": "Added new class"
  }
}
```

Como visualizar informações sobre as alterações em um especificador de confirmação

1. Execute o comando `aws codecommit get-differences`, especificando:
 - O nome do CodeCommit repositório (com a `--repository-name` opção).
 - Os especificadores de confirmação sobre os quais quer informações. Apenas `--after-commit-specifier` é obrigatório. Se você não especificar `--before-commit-specifier`, todos os arquivos atuais do `--after-commit-specifier` serão mostrados.

Por exemplo, para ver informações sobre as diferenças entre os commits com o IDs 317f8570EXAMPLE e 4c925148EXAMPLE em um CodeCommit repositório chamado: MyDemoRepo

```
aws codecommit get-differences --repository-name MyDemoRepo --before-commit-specifier 317f8570EXAMPLE --after-commit-specifier 4c925148EXAMPLE
```

2. Se houver êxito, a saída desse comando incluirá o seguinte:

- Uma lista de diferenças, incluindo o tipo de alteração (A para adicionado, D para excluído ou M para modificado).
- O modo do tipo da alteração do arquivo.
- O ID de objeto do blob do Git que contém a alteração.

Veja um exemplo de saída, com base no comando do exemplo anterior:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3bEXAMPLE",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf28fEXAMPLE",
        "mode": "100644"
      }
    }
  ]
}
```

Como visualizar informações sobre um objeto do blob do Git

1. Execute o comando `aws codecommit get-blob`, especificando:

- O nome do CodeCommit repositório (com a `--repository-name` opção).
- O ID do blob do Git (com a opção `--blob-id`).

Por exemplo, para visualizar informações sobre um blob do Git com o ID de 2eb4af3bEXAMPLE em um CodeCommit repositório chamado: MyDemoRepo

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```


2. Se houver êxito, a saída desse comando incluirá o seguinte:

- O conteúdo codificado em base64 do blob, geralmente um arquivo.

Por exemplo, a saída do comando anterior que pode ser semelhante à seguinte:

```
{
  "content": "QSBCCaW5hcnkgTGFGyToEXAMPLE="
}
```

Visualizar detalhes de confirmação (Git)

Antes de seguir essas etapas, você já deve ter conectado o repositório local ao CodeCommit repositório e confirmado as alterações. Para instruções, consulte [Conexão a um repositório](#).

Para mostrar as alterações na confirmação mais recente de um repositório, execute o comando `git show`.

```
git show
```

O comando gera uma saída semelhante à seguinte:

```
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

Neste e nos exemplos a seguir, commit IDs foi abreviado. A confirmação completa IDs não é exibida.

Para visualizar as alterações ocorridas, use o comando `git show` com o ID da confirmação:

```
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 0000000..080f68f
--- /dev/null
+++ b/horse.txt
@@ -0,0 +1 @@
+The horse (Equus ferus caballus) is one of two extant subspecies of Equus ferus.
```

Para ver as diferenças entre dois commits, execute o `git diff` comando e inclua os dois commits IDs.

```
git diff ce22850d 4f8c6f9d
```

Neste exemplo, a diferença entre as duas confirmações é que dois arquivos foram adicionados. O comando gera uma saída semelhante à seguinte:

```
diff --git a/bees.txt b/bees.txt
new file mode 100644
index 0000000..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 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
```

Para usar o Git para visualizar detalhes sobre as confirmações em um repositório local, execute o comando `git log`:

```
git log
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
commit 94ba1e60
Author: John Doe <johndoe@example.com>
Date:   Mon May 23 15:39:14 2016 -0700

    Added horse.txt

commit 4c925148
Author: Jane Doe <janedoe@example.com>
Date:   Mon May 22 14:54:55 2014 -0700

    Added cat.txt and dog.txt
```

Para mostrar somente IDs confirmações e mensagens, execute o `git log --pretty=oneline` comando:

```
git log --pretty=oneline
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
94ba1e60 Added horse.txt
4c925148 Added cat.txt and dog.txt
```

Para obter mais opções, consulte a documentação do Git.

Compare os commits em AWS CodeCommit

Você pode usar o CodeCommit console para ver as diferenças entre os especificadores de confirmação em um CodeCommit repositório. Você pode visualizar rapidamente a diferença entre uma confirmação e seu pai. Você também pode comparar quaisquer duas referências, incluindo commit IDs.

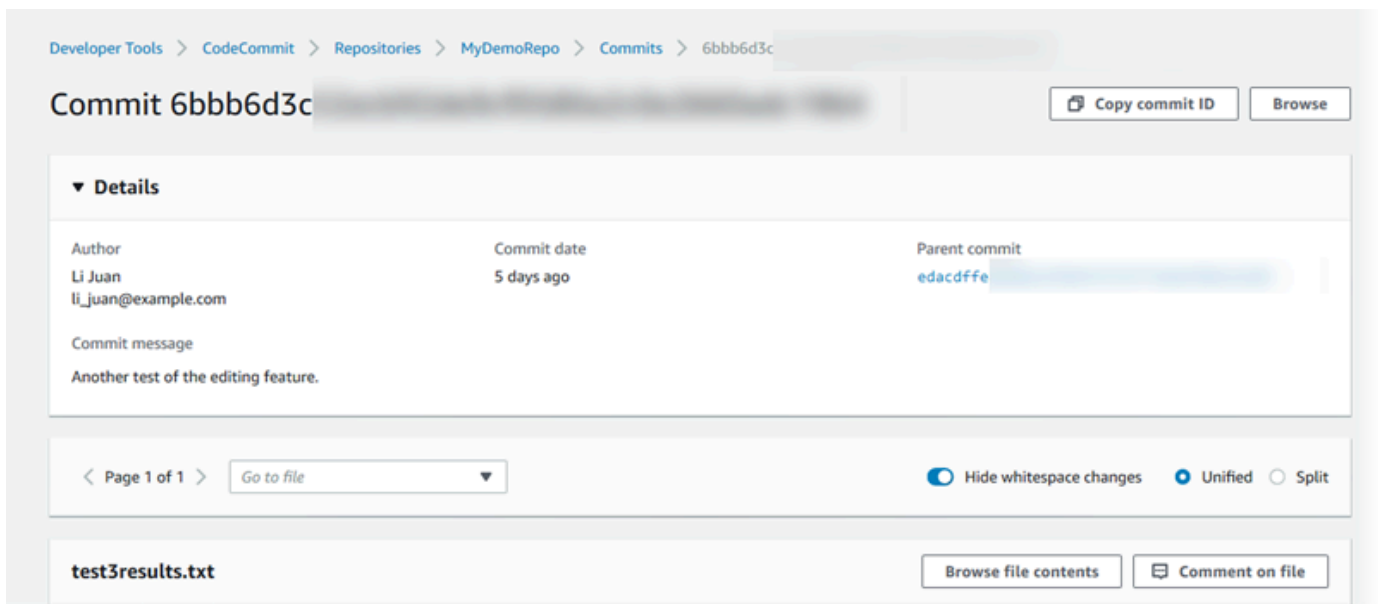
Tópicos

- [Comparar uma confirmação a seu pai](#)
- [Comparação entre dois especificadores de confirmação](#)

Comparar uma confirmação a seu pai

Você pode visualizar rapidamente a diferença entre uma confirmação e seu pai para analisar a mensagem de confirmação, o responsável pela confirmação e o que mudou.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Na página Repositories (Repositórios), escolha o repositório onde você quer visualizar a diferença entre uma confirmação e seu pai.
3. No painel de navegação, selecione Commits.
4. Escolha o ID de confirmação abreviado de qualquer confirmação na lista. Em seguida, visualize as alterações para mostrar os detalhes da confirmação, incluindo as diferenças entre ela e seu pai.



Você pode exibir alterações lado a lado (exibição Split) ou em linha (exibição Unified). Você também pode ocultar ou exibir alterações de espaços em branco. Você também pode adicionar comentários. Para obter mais informações, consulte [Comentar sobre uma confirmação](#).

Note

Suas preferências para visualizar código e outras configurações do console são salvas como cookies do navegador sempre que você as altera. Para obter mais informações, consulte [Trabalhar com preferências do usuário](#).

The screenshot displays the AWS CodeCommit console interface for a specific commit. At the top, a breadcrumb trail shows the navigation path: Developer Tools > CodeCommit > Repositories > MyDemoRepo > Commits. Below this, the commit ID 7d09e44c is shown. The main heading is 'Commit 7d09e44c'. Below the heading are two buttons: 'Copy commit ID' and 'Browse'. A section titled 'Details' is expanded, showing the following information:

Author	Commit date	Parent commit
Mary Major mary_major@example.com	48 minutes ago	e6aca768

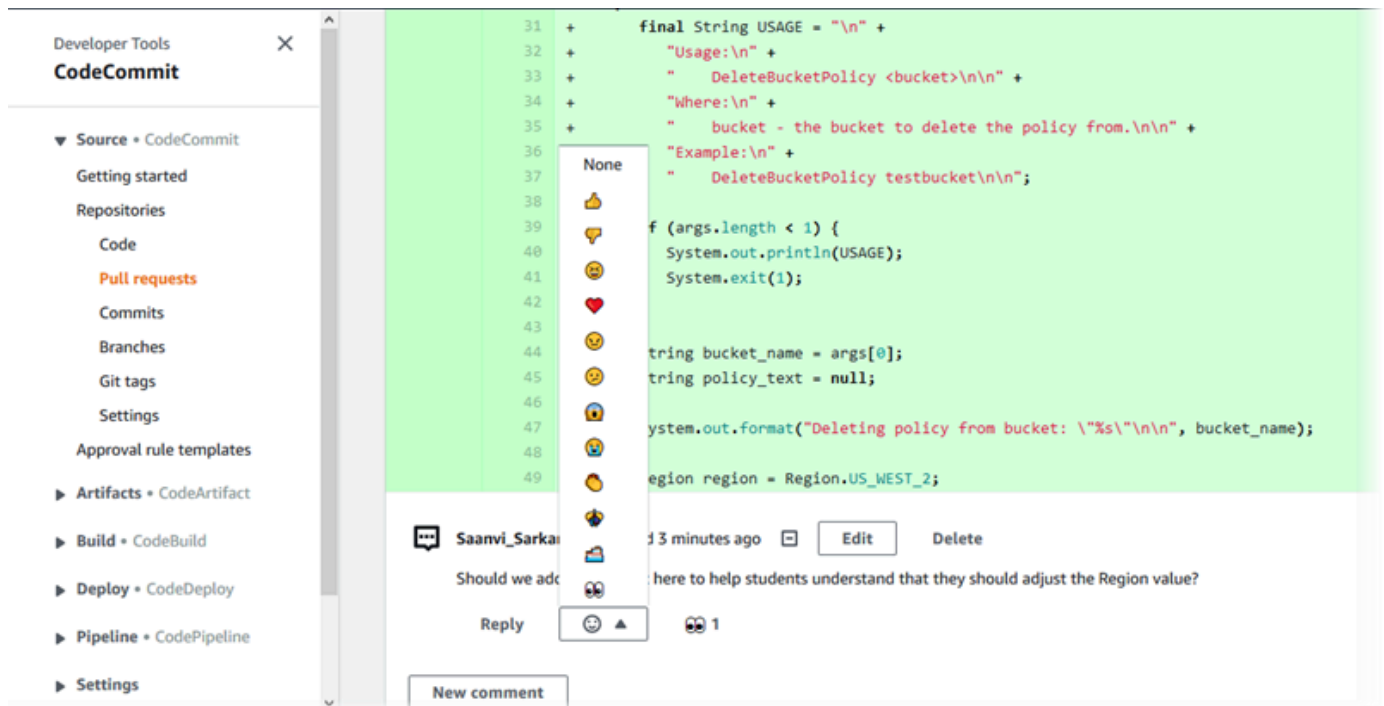
Below the table, the 'Commit message' is displayed: 'Adding a readme file to the repository.'.

At the bottom of the details section, there is a navigation bar with the following elements:

- < Page 1 of 1 >
- Go to file (dropdown menu)
- ☒ Hide whitespace changes
- ☒ Unified
- ☐ Split

The main content area shows the file 'readme.md'. There are two buttons: 'Browse file contents' and 'Comment on file'. The file content is displayed with line numbers and highlights:

```
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

Dependendo do estilo de término da linha, do editor de código e de outros fatores, talvez você veja linhas inteiras adicionadas ou excluídas em vez de alterações específicas em uma linha. O nível de detalhe corresponde ao que é retornado nos comandos `git show` ou `git diff`.

5. Para comparar uma confirmação com seu pai, na guia Commit visualizer (Visualizador de confirmação), escolha o ID de confirmação abreviado. Os detalhes de confirmação, incluindo as alterações entre a confirmação e seu pai, são exibidos.

Comparação entre dois especificadores de confirmação

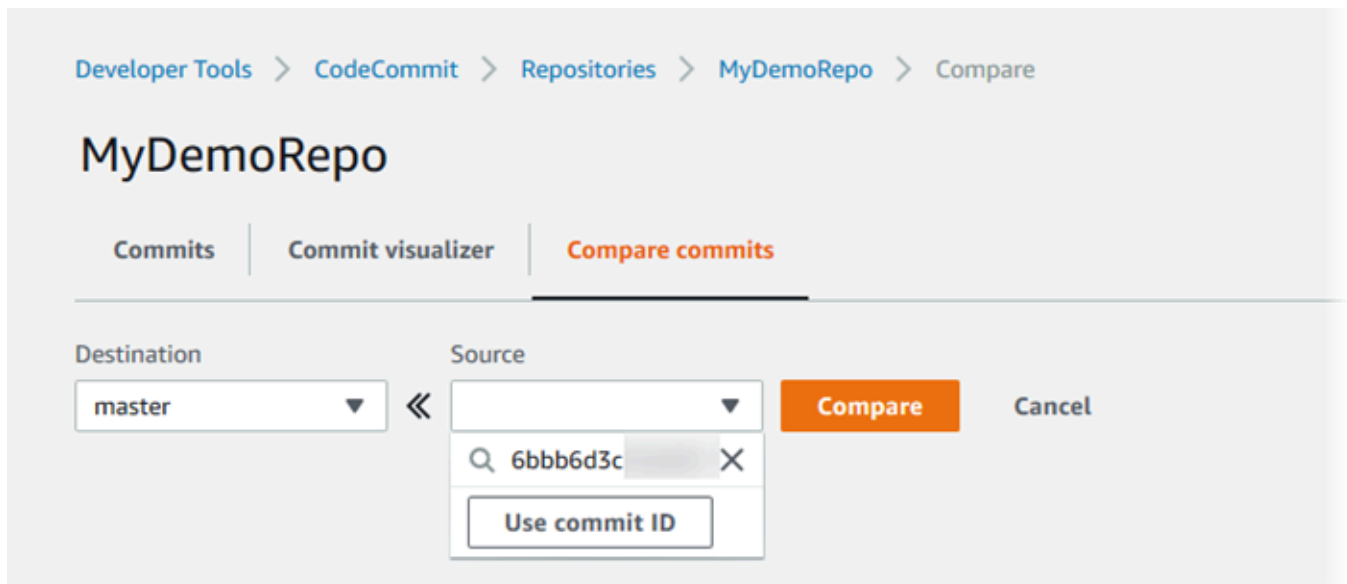
Você pode ver as diferenças entre quaisquer dois especificadores de confirmação no CodeCommit console. Os especificadores de confirmação são referências, como ramificações, tags e confirmação IDs.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Na página Repositories (Repositórios), escolha o repositório onde você quer comparar confirmações, ramificações ou confirmações marcadas com tags.

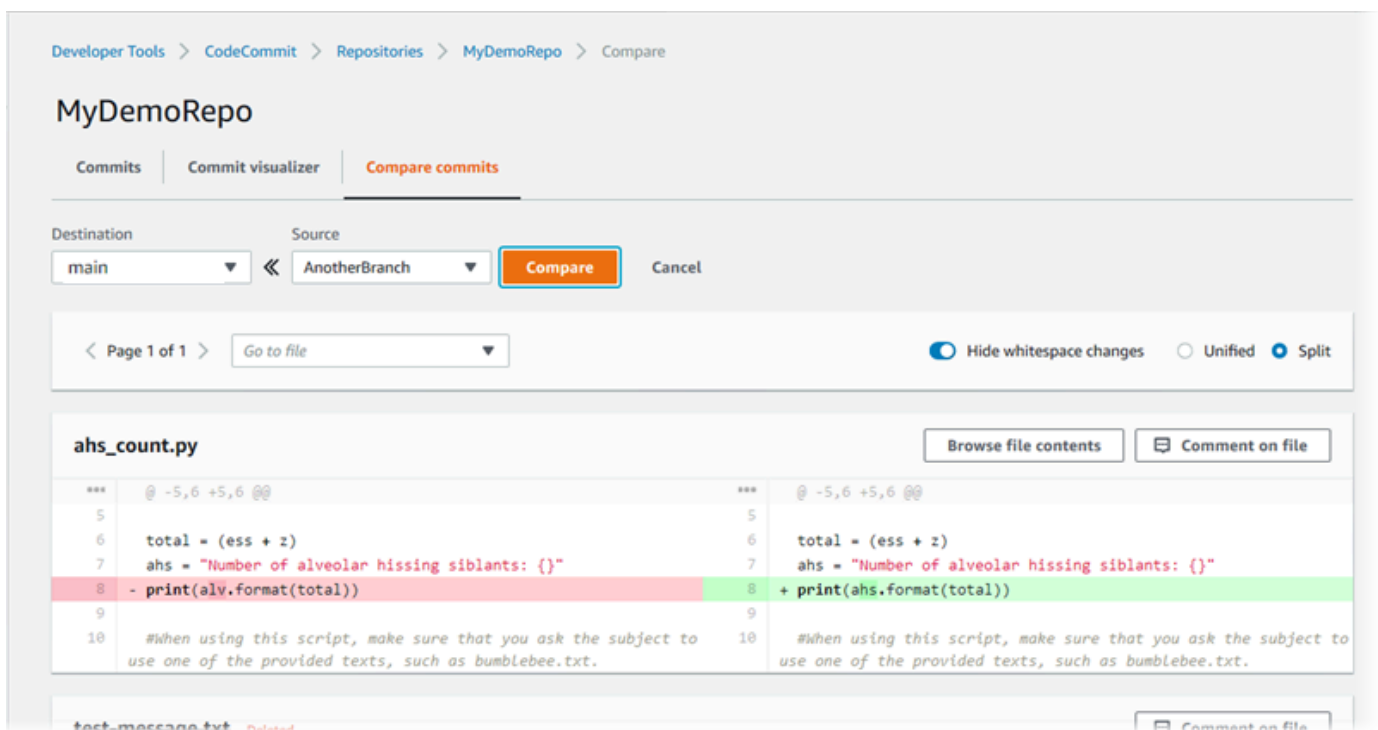
3. No painel de navegação, selecione Commits (Confirmações) e, em seguida, escolha Compare commits (Comparar confirmações).

The screenshot shows the AWS CodeCommit web interface for a repository named 'MyDemoRepo'. The breadcrumb navigation at the top reads: [Developer Tools](#) > [CodeCommit](#) > [Repositories](#) > [MyDemoRepo](#) > [Compare](#). Below the repository name, there are three tabs: 'Commits', 'Commit visualizer', and 'Compare commits' (which is the active tab). Under the 'Compare commits' tab, there are two dropdown menus labeled 'Destination' and 'Source', separated by a double-left arrow icon. To the right of these dropdowns are two buttons: 'Compare' (in orange) and 'Cancel' (in blue).

4. Use as caixas para comparar dois especificadores de confirmação.
 - Para comparar a extremidade da ramificação, escolha o nome da ramificação na lista. Isso seleciona a confirmação mais recente dessa ramificação para comparação.
 - Para comparar uma confirmação a uma tag específica associada a ela, escolha o nome da tag na lista, se houver algum. Isso seleciona a confirmação marcada com a tag para comparação.
 - Para comparar uma confirmação específica, insira ou cole o ID de confirmação na caixa de texto. Para obter o ID completo da confirmação, selecione Commits na barra de navegação e copie o ID da confirmação na lista. Na página Compare commits (Comparar confirmações) cole o ID completo da confirmação na caixa de texto e escolha Use commit ID (Usar ID de confirmação).



5. Após selecionar os especificadores, escolha Compare.



Você pode exibir diferenças lado a lado (exibição Split) ou em linha (exibição Unified). Você também pode ocultar ou exibir alterações de espaços em branco.

6. Para cancelar suas escolhas de comparação, selecione Cancel (Cancelar).

Comente em um commit em AWS CodeCommit

Você pode usar o CodeCommit console para comentar os commits em um repositório e ver e responder aos comentários de outros usuários sobre os commits. Isso pode ajudá-lo a discutir as alterações feitas em um repositório, incluindo:

- Por que as alterações foram feitas.
- Se é necessário fazer mais alterações.
- Se alterações devem ser mescladas em outra ramificação.

Você pode comentar em um compromisso geral, em um arquivo de uma confirmação ou em uma linha específica ou alteração de um arquivo. Também é possível vincular a uma linha de código selecionando a linha e copiando a URL resultante no navegador.

Note

Para obter melhores resultados, use comentários quando estiver conectado como um usuário do IAM. A funcionalidade de comentários não é otimizada para usuários que fazem login com credenciais de conta raiz, acesso federado ou credenciais temporárias.

Tópicos

- [Visualizar comentários em uma confirmação em um repositório](#)
- [Adicionar e responder a comentários em uma confirmação em um repositório](#)
- [Visualizar, adicionar, atualizar e responder a comentários \(AWS CLI\)](#)

Visualizar comentários em uma confirmação em um repositório

Você pode usar o CodeCommit console para ver os comentários em um commit.

Como visualizar comentários em uma confirmação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório do qual você deseja rever os comentários em confirmações.

3. No painel de navegação, selecione Commits. Escolha o ID da confirmação em que você deseja visualizar comentários.

A página para essa confirmação é exibida, junto com quaisquer comentários.

Adicionar e responder a comentários em uma confirmação em um repositório

Você pode usar o CodeCommit console para adicionar comentários à comparação de um commit com um pai ou à comparação entre dois commits especificados. Você também pode responder aos comentários com emojis, com seus próprios comentários ou ambos.

Adicionar e responder a comentários em uma confirmação (console)

Você pode adicionar e responder a comentários em uma confirmação com texto e emojis. Seus comentários e emojis são marcados como pertencentes ao usuário ou perfil do IAM que você usou para entrar no console.

Para adicionar e responder a comentários em uma confirmação

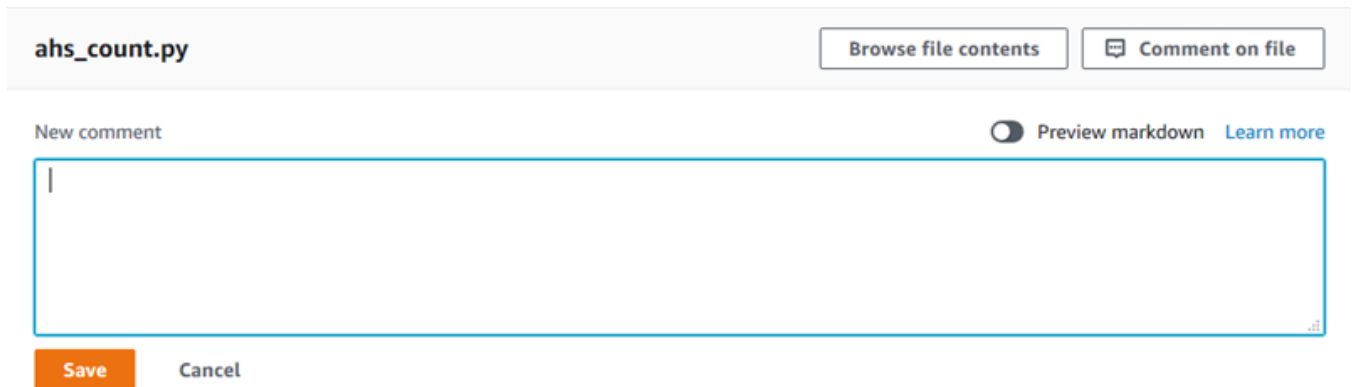
1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde deseja comentar nas confirmações.
3. No painel de navegação, selecione Commits. Escolha o ID da confirmação em que você deseja adicionar ou responder aos comentários.

A página para essa confirmação é exibida, junto com quaisquer comentários.

4. Para adicionar um comentário, siga um destes procedimentos:
 - Para adicionar um comentário geral, em Comments on changes (Comentários em alterações), insira seu comentário e, em seguida, escolha Save (Salvar). Você pode usar [Markdown](#) ou inserir seu comentário em texto simples.



- Para adicionar um comentário a um arquivo na confirmação, encontre o nome do arquivo. Escolha Comment on file (Comentar no arquivo), insira seus comentários e, em seguida, escolha Save (Salvar).



- Para adicionar um comentário a uma linha alterada na confirmação, vá até a linha onde a alteração aparece. Escolha a bolha de comentário



insira o comentário e, em seguida, escolha Save (Salvar).

ahs_count.py [Browse file contents](#) [Comment on file](#)

```
*** @ -4,7 +4,7 @@
4  z = z.count('z')
5
6  total = (ess + z)
7  - alv = "Number of alveolar hissing sibilants: { }"
   + ahs = "Number of alveolar hissing sibilants: { }"

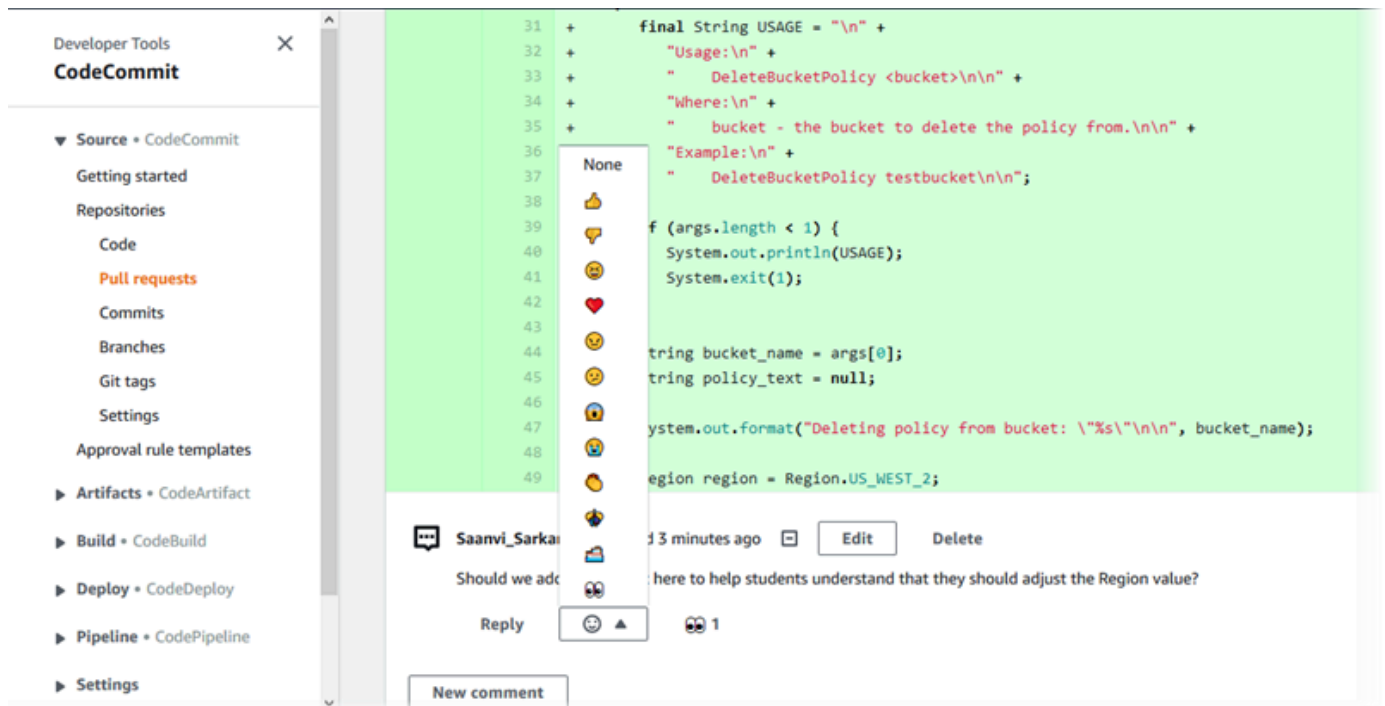
New comment ☒ Preview markdown Learn more
You've reverted to the old value here, which won't work. This
should remain alv.
Save Cancel

8  print(alv.format(total))
8  print(ahs.format(total))
```

Note

Você pode editar seu comentário depois de salvá-lo. Você também pode excluir seu conteúdo. O comentário permanecerá com uma mensagem informando que o conteúdo foi excluído. Considere usar o modo Preview markdown (Visualização de markdown) para o seu comentário antes de salvá-lo.

5. Para responder a comentários em uma confirmação, escolha Reply (Responder). Para responder a um comentário com um emoji, escolha o emoji que você quer na lista. Você só pode escolher um emoji por comentário. Se você quiser alterar sua reação de emoji, escolha uma reação diferente na lista ou escolha Nenhuma para remover sua reação.

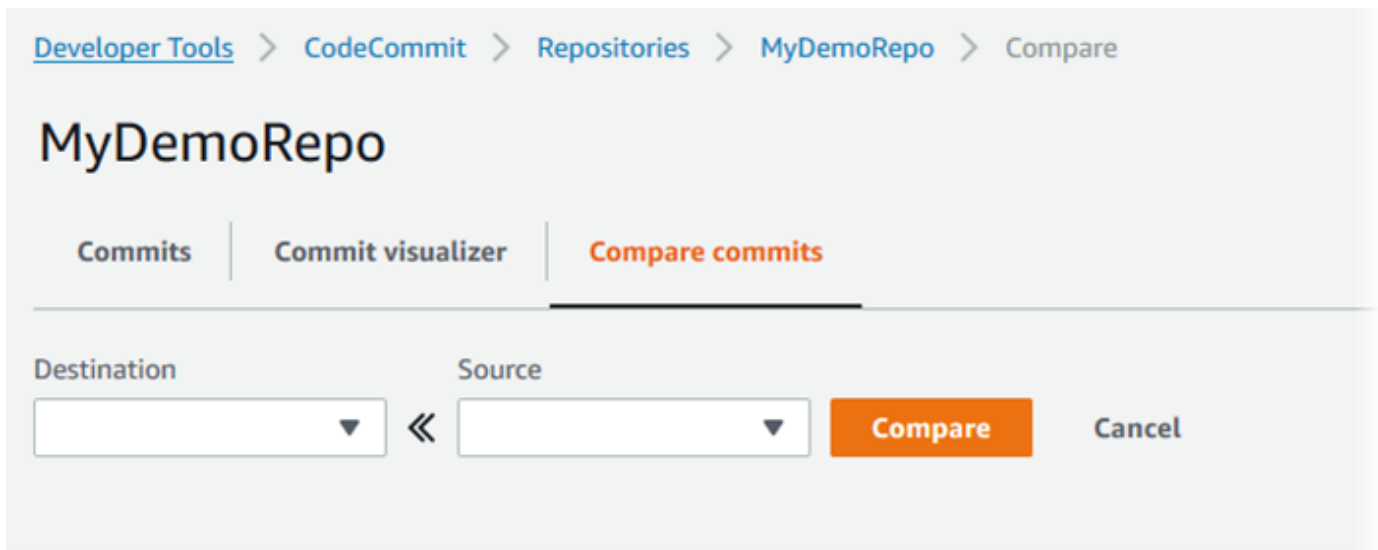


Adicionar e responder a comentários ao comparar dois especificadores de confirmação

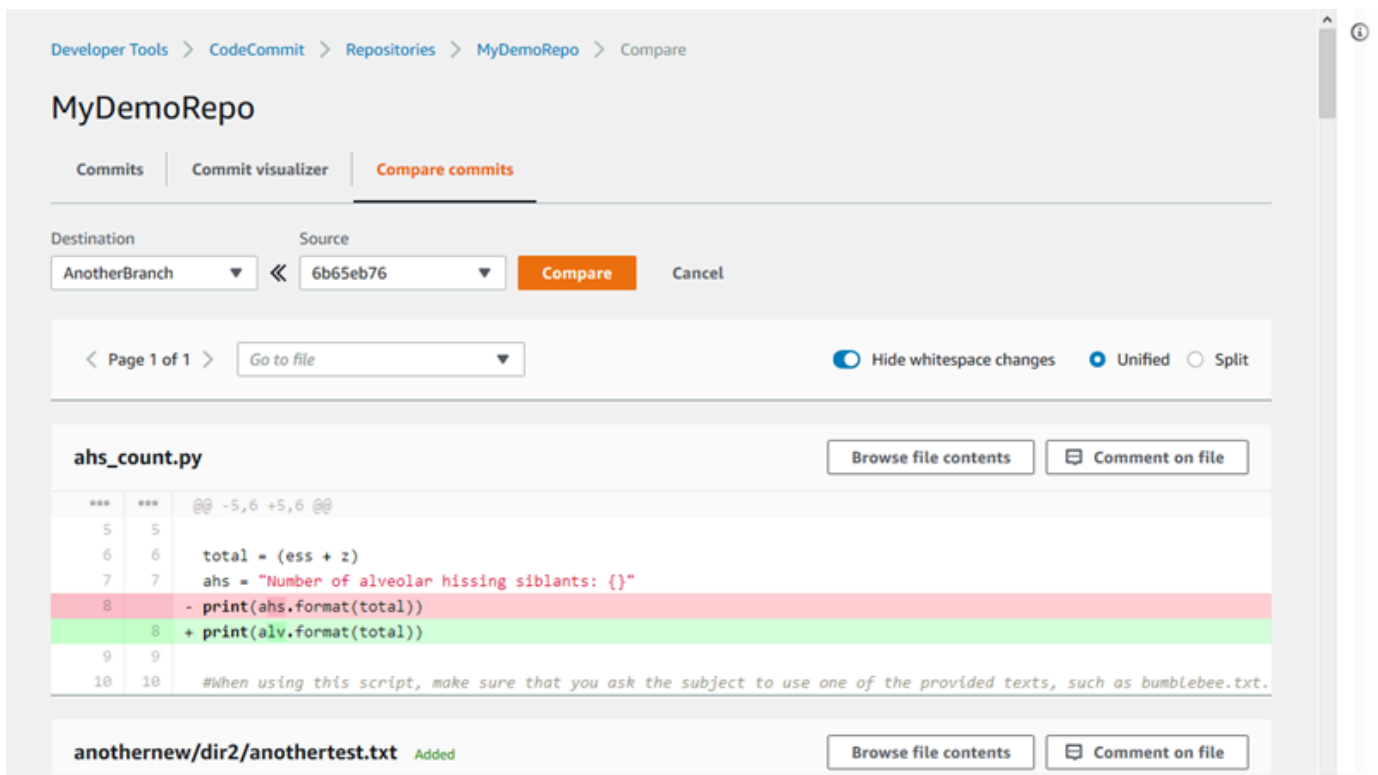
Você pode adicionar comentários a uma comparação entre ramificações, tags ou confirmações.

Para adicionar ou responder a comentários ao comparar especificadores de confirmação

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o repositório onde você quer comparar confirmações, ramificações ou confirmações marcadas com tags.
3. No painel de navegação, selecione Commits (Confirmações) e, em seguida, escolha a guia Compare commits (Comparar confirmações).



4. Use os campos Destination (Destino) e Source (Origem) para comprar dois especificadores de confirmação. Use as listas suspensas ou cole no commit. IDs Escolha Compare.



5. Siga um dos procedimentos abaixo:

- Para adicionar comentários a arquivos ou linhas, escolha a bolha de comentários



- Para adicionar comentários gerais sobre as alterações comparadas, acesse `Comments on changes`.

Visualizar, adicionar, atualizar e responder a comentários (AWS CLI)

Você pode visualizar, adicionar, responder, atualizar e excluir o conteúdo de um comentário executando os seguintes comandos:

- Para visualizar os comentários na comparação entre duas confirmações, execute [get-comments-for-compared-commit](#).
- Para visualizar detalhes em um comentário, execute [get-comment](#).
- Para excluir o conteúdo de um comentário que você criou, execute [delete-comment-content](#).
- Para criar um comentário na comparação entre duas confirmações, execute [post-comment-for-compared-commit](#).
- Para atualizar um comentário, execute [update-comment](#).
- Para responder a um comentário, execute [post-comment-reply](#).
- Para responder a um comentário com um emoji, execute [put-comment-reaction](#).
- Para ver as reações de emoji a um comentário, execute [get-comment-reactions](#).

Como visualizar comentários em uma confirmação

1. Execute o comando `get-comments-for-compared-commit`, especificando:
 - O nome do CodeCommit repositório (com a `--repository-name` opção).
 - O ID de confirmação completo da confirmação "depois" para estabelecer a direção da comparação (com `--after-commit-id` option).
 - O ID de confirmação completo da confirmação "antes" para estabelecer a direção da comparação (com a opção `--before-commit-id`).
 - (Opcional) Um token de enumeração para retornar o lote seguintes de resultados (com a opção `--next-token`).
 - (Opcional) Um inteiro não negativo no número de resultados retornados (com a opção `--max-results`).

Por exemplo, para visualizar comentários realizados na comparação entre duas confirmações em um repositório chamado *MyDemoRepo*:

```
aws codecommit get-comments-for-compared-commit --repository-name MyDemoRepo --  
before-commit-id 6e147360EXAMPLE --after-commit-id 317f8570EXAMPLE
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "commentsForComparedCommitData": [  
    {  
      "afterBlobId": "1f330709EXAMPLE",  
      "afterCommitId": "317f8570EXAMPLE",  
      "beforeBlobId": "80906a4cEXAMPLE",  
      "beforeCommitId": "6e147360EXAMPLE",  
      "comments": [  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "123Example",  
          "commentId": "ff30b348EXAMPLEb9aa670f",  
          "content": "Whoops - I meant to add this comment to the line, not  
the file, but I don't see how to delete it.",  
          "creationDate": 1508369768.142,  
          "deleted": false,  
          "CommentId": "123abc-EXAMPLE",  
          "lastModifiedDate": 1508369842.278,  
          "callerReactions": [],  
          "reactionCounts":  
            {  
              "SMILE" : 6,  
              "THUMBSUP" : 1  
            }  
        },  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "123Example",  
          "commentId": "553b509bEXAMPLE56198325",  
          "content": "Can you add a test case for this?",  
          "creationDate": 1508369612.240,  
          "deleted": false,  
          "commentId": "456def-EXAMPLE",
```



```

        "lastModifiedDate": 1508369612.240,
        "callerReactions": [],
        "reactionCounts":
        {
            "THUMBSUP" : 2
        }
    },
    "location": {
        "filePath": "cl_sample.js",
        "filePosition": 1232,
        "relativeFileVersion": "after"
    },
    "repositoryName": "MyDemoRepo"
}
],
"nextToken": "exampleToken"
}

```

Para visualizar detalhes de um comentário em uma confirmação

1. Execute o comando `get-comment`, especificando o ID de comentário gerado pelo sistema. Por exemplo:

```
aws codecommit get-comment --comment-id ff30b348EXAMPLEb9aa670f
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```

{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "123Example",
    "commentId": "ff30b348EXAMPLEb9aa670f",
    "content": "Whoops - I meant to add this comment to the line, but I don't see
how to delete it.",
    "creationDate": 1508369768.142,
    "deleted": false,
    "commentId": "",
    "lastModifiedDate": 1508369842.278,
    "callerReactions": [],
    "reactionCounts":
    {

```

```
        "SMILE" : 6,  
        "THUMBSUP" : 1  
    }  
}  
}
```

Para excluir o conteúdo de um comentário em uma confirmação

1. Execute o comando `delete-comment-content`, especificando o ID de comentário gerado pelo sistema. Por exemplo:

```
aws codecommit delete-comment-content --comment-id ff30b348EXAMPLEb9aa670f
```

Note

Você só pode excluir o conteúdo de um comentário se tiver a `AWSCodeCommitFullAccess` política aplicada ou se tiver a `DeleteCommentContent` permissão definida como Permitir.

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "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  
      }  
  }  
}
```

Para criar um comentário em uma confirmação

1. Execute o comando `post-comment-for-compared-commit`, especificando:
 - O nome do CodeCommit repositório (com a `--repository-name` opção).
 - O ID de confirmação completo da confirmação "depois" para estabelecer a direção da comparação (com a opção `--after-commit-id`).
 - O ID de confirmação completo da confirmação "antes" para estabelecer a direção da comparação (com a opção `--before-commit-id`).
 - Um token de idempotência exclusivo gerado pelo cliente (com a opção `--client-request-token`).
 - O conteúdo do seu comentário (com a opção `--content`).
 - Uma lista de informações de localização sobre onde colocar o comentário, incluindo:
 - O nome do arquivo que está sendo comparado, incluindo sua extensão e subdiretório, se houver (com o atributo `filePath`).
 - O número da linha da alteração dentro de um arquivo comparado (com o atributo `filePosition`).
 - Se o comentário sobre a alteração estiver "antes" ou "depois" na comparação entre as ramificações de origem e destino (com o atributo `relativeFileVersion`).

Por exemplo, para adicionar o comentário *"Can you add a test case for this?"* sobre a alteração no *cl_sample.js* arquivo na comparação entre dois commits em um repositório chamado: *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
  "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"
}

```

Para atualizar um comentário em uma confirmação

1. Execute o comando `update-comment` especificando o ID de comentário gerado pelo sistema e o conteúdo para substituir o conteúdo existente.

Por exemplo, para adicionar o conteúdo *"Fixed as requested. I'll update the pull request."* a um comentário com um ID de *442b498bEXAMPLE5756813*:

```
aws codecommit update-comment --comment-id 442b498bEXAMPLE5756813 --content "Fixed
as requested. I'll update the pull request."
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```

{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "",
    "commentId": "442b498bEXAMPLE5756813",
    "content": "Fixed as requested. I'll update the pull request.",
    "creationDate": 1508369929.783,
    "deleted": false,
    "lastModifiedDate": 1508369929.287,
    "callerReactions": [],

```

```
    "reactionCounts":  
      {  
        "THUMBSUP" : 2  
      }  
  }  
}
```

Para responder a um comentário em uma confirmação

1. Para enviar uma resposta a um comentário em uma solicitação pull, execute o comando `post-comment-reply`, especificando:
 - O ID gerado pelo sistema do comentário ao qual você deseja responder (com a opção `--in-reply-to`).
 - Um token de idempotência exclusivo gerado pelo cliente (com a opção `--client-request-token`).
 - O conteúdo da sua resposta (com a opção `--content`).

Por exemplo, para adicionar a resposta *"Good catch. I'll remove them."* ao comentário com o ID gerado pelo sistema de: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit post-comment-reply --in-reply-to abcd1234EXAMPLEb5678efgh --  
content "Good catch. I'll remove them." --client-request-token 123Example
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "123Example",  
    "commentId": "442b498bEXAMPLE5756813",  
    "content": "Good catch. I'll remove them.",  
    "creationDate": 1508369829.136,  
    "deleted": false,  
    "CommentId": "abcd1234EXAMPLEb5678efgh",  
    "lastModifiedDate": 150836912.221,  
    "callerReactions": [],  
    "reactionCounts": []  
  }  
}
```

Para responder a um comentário em uma confirmação com um emoji

1. Para responder a um comentário em uma solicitação pull com um emoji ou para alterar o valor da reação do emoji, execute o comando `put-comment-reaction`, especificando:
 - O ID gerado pelo sistema do comentário ao qual você deseja responder com um emoji.
 - O valor da reação que você deseja adicionar ou atualizar. Os valores aceitáveis incluem emojis, códigos curtos e valores Unicode compatíveis.

Os valores a seguir são compatíveis com emojis em: 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
	nenhuma	U+0000

Por exemplo, para adicionar o emoji *:thumbsup:* ao comentário com o ID gerado pelo sistema de: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit put-comment-reaction --comment-id abcd1234EXAMPLEb5678efgh --  
reaction-value :thumbsup:
```

2. Se for bem-sucedido, esse comando não gerará nenhum resultado.

Para ver as reações de emoji a um comentário

1. Para ver as reações de emojis a um comentário, incluindo os usuários que reagiram com esses emojis, execute o comando `get-comment-reactions`, especificando o ID do comentário gerado pelo sistema.

Por exemplo, para ver as reações de emoji ao comentário com o ID gerado pelo sistema de: *abcd1234EXAMPLEb5678efgh*

```
aws codecommit get-comment-reactions --comment-id abcd1234EXAMPLEb5678efgh
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "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"
      ]
    }
  ]
}
```

Crie uma tag Git em AWS CodeCommit

É possível usar uma tag do Git para marcar uma confirmação com um rótulo que ajuda outros usuários do repositório a entender sua importância. Para criar uma tag Git em um CodeCommit repositório, você pode usar o Git de um repositório local conectado ao repositório. CodeCommit Depois de criar uma tag Git no repositório local, você pode usá-la `git push --tags` para enviá-la para o repositório. CodeCommit

Para obter mais informações, consulte [Visualizar detalhes de tags](#).

Usar o Git para criar uma tag

Siga estas etapas para usar o Git de um repositório local para criar uma tag do Git em um repositório. CodeCommit

Nessas etapas, presumimos que você já tenha conectado o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

1. Execute o `git tag new-tag-name commit-id` comando, onde **new-tag-name** está o nome da nova tag Git e **commit-id** o ID do commit a ser associado à tag Git.

Por exemplo, o seguinte comando cria uma tag do Git chamada beta e a associa ao ID de confirmação dc082f9a...af873b88:

```
git tag beta dc082f9a...af873b88
```

2. Para enviar a nova tag Git do repositório local para o CodeCommit repositório, execute o `git push remote-name new-tag-name` comando, onde **remote-name** está o nome do CodeCommit repositório e **new-tag-name** o nome da nova tag Git.

Por exemplo, para enviar uma nova tag do Git chamada beta para um CodeCommit repositório chamado: origin

```
git push origin beta
```

Note

Para enviar todas as novas tags Git do seu repositório local para o CodeCommit repositório, execute. `git push --tags`

Para garantir que seu repositório local seja atualizado com todas as tags Git no repositório, CodeCommit `git fetch` execute e siga por. `git fetch --tags`

Para obter mais opções, consulte a documentação do Git.

Veja os detalhes da tag Git em AWS CodeCommit

No Git, uma tag é um rótulo que você pode aplicar a uma referência, como uma confirmação, para marcá-la com informações que podem ser importantes para outros usuários do repositório. Por exemplo, você pode marcar a confirmação que foi o ponto da versão beta de um projeto com a tag **beta**. Para obter mais informações, consulte [Usar o Git para criar uma tag](#). As tags do Git são diferentes das tags de repositório. Para obter mais informações sobre como usar tags de repositório, consulte [Adicionar uma tag a um repositório](#).

Você pode usar o AWS CodeCommit console para visualizar informações sobre as tags do Git no seu repositório, incluindo a data e a mensagem de confirmação da confirmação referenciada por cada tag do Git. No console, você pode comparar a confirmação referenciada pela tag com o

cabeçalho da ramificação padrão do seu repositório. Como qualquer outra confirmação, também é possível visualizar o código no ponto dessa tag do Git.

Também é possível usar o Git em seu terminal ou na linha de comando para visualizar detalhes sobre tags do Git em um repositório local.

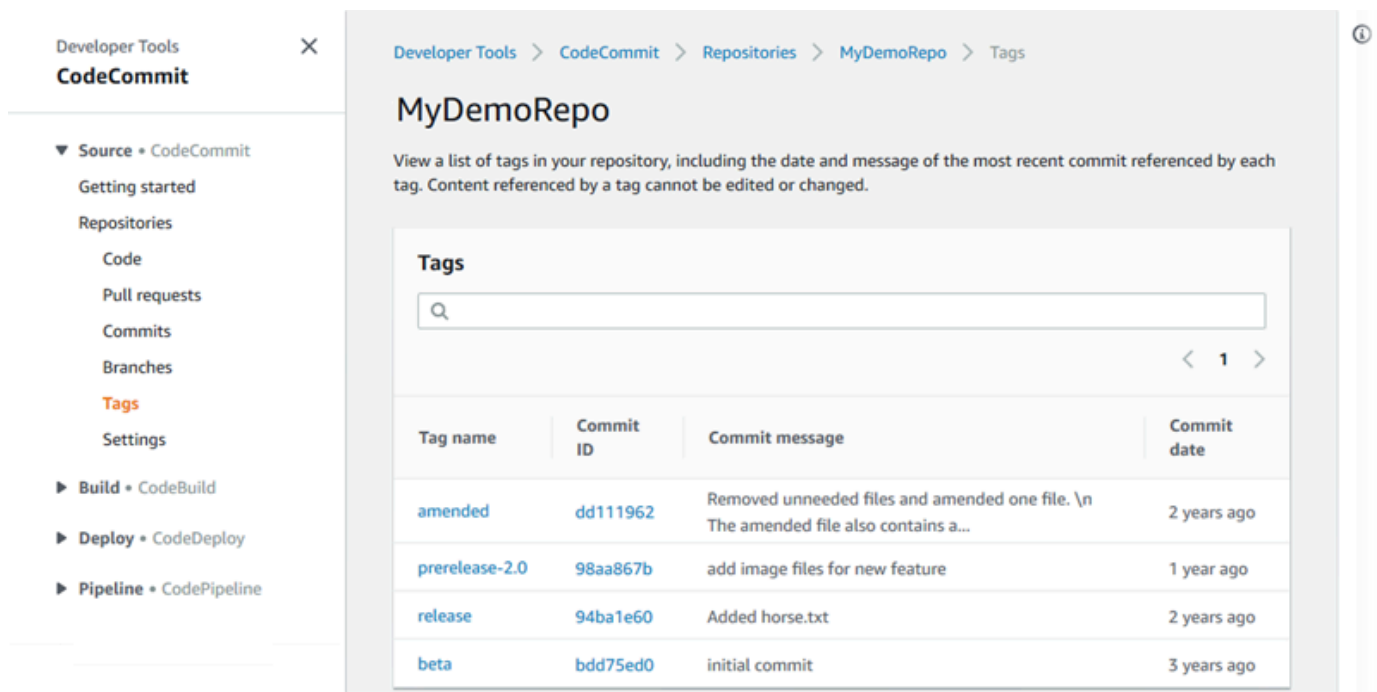
Tópicos

- [Visualizar detalhes de tags \(console\)](#)
- [Visualizar detalhes de tags do Git \(Git\)](#)

Visualizar detalhes de tags (console)

Use o AWS CodeCommit console para ver rapidamente uma lista de tags do Git para seu repositório e detalhes sobre os commits referenciados pelas tags do Git.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja visualizar as tags.
3. No painel de navegação, selecione Git tags (Tags do Git).



The screenshot shows the AWS CodeCommit console interface. On the left is a navigation pane with a sidebar menu. The main content area displays the 'Tags' page for a repository named 'MyDemoRepo'. The page includes a search bar, a table of tags, and a description of the tags page.

Developer Tools **CodeCommit**

▼ Source • CodeCommit

- Getting started
- Repositories
- Code
- Pull requests
- Commits
- Branches
- Tags**
- Settings

► Build • CodeBuild

► Deploy • CodeDeploy

► Pipeline • CodePipeline

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Tags

MyDemoRepo

View a list of tags in your repository, including the date and message of the most recent commit referenced by each tag. Content referenced by a tag cannot be edited or changed.

Tags

Search:

< 1 >

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. Execute um destes procedimentos:
 - Para visualizar o código como ele estava nessa confirmação, selecione o nome da tag do Git.

- Para visualizar detalhes da confirmação, incluindo a mensagem de confirmação completa, o responsável pela confirmação e o autor, escolha o ID abreviado da confirmação.

Visualizar detalhes de tags do Git (Git)

Para usar o Git a fim de visualizar detalhes sobre tags do Git em um repositório local, execute um dos seguintes comandos:

- [git tag](#) para visualizar uma lista de nomes de tags do Git.
- [git show](#) para visualizar informações sobre uma tag do Git específica.
- [git ls-remote](#) para visualizar informações sobre as tags do Git em um repositório. CodeCommit

Note

Para garantir que seu repositório local seja atualizado com todas as tags Git no CodeCommit repositório, git fetch execute e siga por. git fetch --tags

Nas etapas a seguir, presumimos que você já tenha conectado o repositório local a um CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

Como visualizar uma lista de tags do Git em um repositório local

1. Execute o comando git tag:

```
git tag
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
beta  
release
```

Note

Se nenhuma tiver sido definida, git tag não retornará nada.

Para obter mais opções, consulte a documentação do Git.

Para visualizar informações sobre uma tag do Git em um repositório local

1. Execute o comando `git show tag-name`. Por exemplo, para visualizar informações sobre uma tag do Git chamada beta, execute:

```
git show beta
```

2. Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
commit 317f8570...ad9e3c09
Author: John Doe <johndoe@example.com>
Date:   Tue Sep 23 13:49:51 2014 -0700

    Added horse.txt

diff --git a/horse.txt b/horse.txt
new file mode 100644
index 0000000..df42ff1
--- /dev/null
+++ b/horse.txt
@@ -0,0 +1 @@
+The horse (Equus ferus caballus) is one of two extant subspecies of Equus ferus
\ No newline at end of file
```

Note

Para sair da saída das informações da tag do Git, digite `:q`.

Para obter mais opções, consulte a documentação do Git.

Para visualizar informações sobre tags Git em um repositório CodeCommit

1. Execute o comando `git ls-remote --tags`.

```
git ls-remote --tags
```

2. Se for bem-sucedido, esse comando produzirá como saída uma lista das tags Git no CodeCommit repositório:

```
129ce87a...70fbffba    refs/tags/beta
785de9bd...59b402d8    refs/tags/release
```

Se nenhuma tag do Git tiver sido definida, `git ls-remote --tags` retornará uma linha em branco.

Para obter mais opções, consulte a documentação do Git.

Excluir uma tag Git em AWS CodeCommit

Para excluir uma tag do Git em um CodeCommit repositório, use o Git de um repositório local conectado ao repositório. CodeCommit

Usar o Git para excluir uma tag do Git

Siga estas etapas para usar o Git de um repositório local para excluir uma tag do Git em um repositório. CodeCommit

Essas etapas são escritas com a suposição de que você já conectou o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

1. Para excluir a tag Git do repositório local, execute o `git tag -d tag-name` comando onde **tag-name** está o nome da tag Git que você deseja excluir.

Tip

Para obter uma lista de nomes de tags do Git, execute `git tag`.

Por exemplo, para excluir uma tag do Git no repositório local chamado beta:

```
git tag -d beta
```

2. Para excluir a tag Git do CodeCommit repositório, execute o `git push remote-name --delete tag-name` comando onde **remote-name** está o apelido que o repositório local usa para o CodeCommit repositório e é **tag-name** o nome da tag Git que você deseja excluir do repositório. CodeCommit

 Tip

Para obter uma lista dos nomes dos CodeCommit repositórios e seus URLs, execute o `git remote -v` comando.

Por exemplo, para excluir uma tag do Git chamada `beta` no CodeCommit repositório com o nome: `origin`

```
git push origin --delete beta
```

Trabalhando com ramificações em AWS CodeCommit repositórios

O que é uma ramificação? No Git, as ramificações são simplesmente ponteiros ou referências a uma confirmação. Durante o desenvolvimento, elas são uma forma conveniente de organizar seu trabalho. As ramificações podem ser usadas para separar o trabalho em uma versão nova ou diferente de arquivos sem afetar o trabalho em outras ramificações. Elas também podem ser usadas para desenvolver novos recursos, armazenar uma versão específica do seu projeto a partir de uma confirmação particular e muito mais. Quando você cria sua primeira confirmação, uma ramificação padrão é criada para você. Essa ramificação padrão é aquela usada como base ou ramificação padrão em repositórios locais quando os usuários clonam o repositório. O nome dessa ramificação padrão varia dependendo de como você cria sua primeira confirmação. Se você adicionar o primeiro arquivo ao seu repositório usando o CodeCommit console AWS CLI, o ou um dos SDKs, o nome dessa ramificação padrão será main. Esse é o nome usado para a ramificação padrão em todos os exemplos desse guia. Se você enviar sua primeira confirmação usando um cliente Git, o nome da ramificação padrão é o que o cliente Git especifica como padrão. Considere configurar seu cliente Git para usar principal como nome da ramificação inicial.

Em CodeCommit, você pode alterar a ramificação padrão do seu repositório. Você também pode criar e excluir ramificações e visualizar detalhes sobre uma ramificação. Você pode comparar rapidamente as diferenças entre uma ramificação e a ramificação padrão (ou duas ramificações). Para visualizar o histórico de ramificações e mesclagens no seu repositório, você pode usar o [Visualizador de confirmações](#), mostrado no gráfico a seguir.

Developer Tools **CodeCommit**

▼ Source • CodeCommit

- Getting started
- Repositories
- Code
- Pull requests
- Commits**
- Branches
- Tags
- Settings

► Build • CodeBuild

► Deploy • CodeDeploy

► Pipeline • CodePipeline

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Commits

MyDemoRepo

Commits | **Commit visualizer** | Compare commits

Commit visualizer

Commit Hash	Message	Time Ago
d615e7ae	Merge branch 'AnotherBranch' into testbranch	2 minutes ago
b6589863	Added another file.	2 minutes ago
73a6e39c	remote-tracking branch 'refs/remotes/origin/jane-branch' into jane-branch	
6bbb6d3c	Another test of the editing feature.	20 minutes ago
edacdfef	Testing this out to see how well it works.	23 minutes ago
70bb94d7	Revised test results with correct information.	36 minutes ago
b78e6d1c	Merge branch 'results' into testbranch	50 minutes ago
84b7d158	Edited ahs_count.py	50 minutes ago

Para obter informações sobre como trabalhar com outros aspectos do seu repositório em CodeCommit [Trabalhar com repositórios](#), consulte [Trabalhar com arquivos](#), [Trabalhar com solicitações pull](#), [Trabalhar com confirmações](#), e [Trabalhar com preferências do usuário](#)

Tópicos

- [Crie uma filial em AWS CodeCommit](#)
- [Limite os envios e fusões às ramificações em AWS CodeCommit](#)
- [Veja os detalhes da filial em AWS CodeCommit](#)
- [Compare e mescle filiais em AWS CodeCommit](#)
- [Alterar as configurações da ramificação em AWS CodeCommit](#)
- [Excluir uma ramificação em AWS CodeCommit](#)

Crie uma filial em AWS CodeCommit

Você pode usar o CodeCommit console ou o AWS CLI para criar branches para seu repositório. Esta é uma maneira rápida de separar o trabalho em uma versão nova ou diferente de arquivos sem afetar o trabalho na ramificação padrão. Depois de criar uma ramificação no CodeCommit console, você deve transferir essa alteração para seu repositório local. Como alternativa, você pode criar uma ramificação localmente e, em seguida, usar o Git de um repositório local conectado ao CodeCommit repositório para enviar essa alteração.

Tópicos

- [Criar uma ramificação \(console\)](#)
- [Criar uma ramificação \(Git\)](#)
- [Criar uma ramificação \(AWS CLI\)](#)

Criar uma ramificação (console)

Você pode usar o CodeCommit console para criar uma ramificação em um CodeCommit repositório. Da próxima vez que os usuários extraírem alterações do repositório, eles verão a nova ramificação.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja criar uma ramificação.
3. No painel de navegação, escolha Branches.
4. Escolha Create branch.

Create branch

Branch name
feature_advanced-class

Branch from

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

Em Branch name (Nome da ramificação), insira um nome para a ramificação. Em Branch from (Ramificação de), escolha uma ramificação ou tag da lista ou cole um ID de confirmação. Escolha Create branch.

Criar uma ramificação (Git)

Siga estas etapas para usar o Git de um repositório local para criar uma ramificação em um repositório local e, em seguida, enviar essa ramificação para o repositório. CodeCommit

Essas etapas são escritas com a suposição de que você já conectou o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

1. Crie uma ramificação em seu repositório local executando o git checkout -b **new-branch-name** comando, onde **new-branch-name** está o nome da nova ramificação.

Por exemplo, o seguinte comando cria uma ramificação chamada MyNewBranch no repositório local:

```
git checkout -b MyNewBranch
```

2. Para enviar a nova ramificação do repositório local para o CodeCommit repositório, execute o git push comando, especificando o e o. **remote-name new-branch-name**

Por exemplo, para enviar uma nova ramificação no repositório local chamada MyNewBranch para o CodeCommit repositório com o apelido: origin

```
git push origin MyNewBranch
```

Note

Se você adicionar a -u opção a git push (por exemplo, git push -u origin main), no futuro poderá executar git push sem **remote-name branch-name**. As informações de rastreamento upstream estão definidas. Para obter as informações de rastreamento upstream, execute git remote show **remote-name** (por exemplo, git remote show origin). Para ver uma lista de todas as ramificações de rastreamento local e remoto, execute git branch --all.

Para configurar uma ramificação no repositório local conectada a uma ramificação no CodeCommit repositório, execute. git checkout **remote-branch-name**

Para obter mais opções, consulte a documentação do Git.

Criar uma ramificação (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Siga estas etapas para usar o AWS CLI para criar uma ramificação em um CodeCommit repositório e, em seguida, enviar essa ramificação para o CodeCommit repositório. Para ver as etapas para criar uma confirmação inicial e especificar o nome da ramificação padrão para um repositório vazio, consulte [Criar a primeira confirmação para um repositório usando a AWS CLI](#).

1. Execute o comando `create-branch`, especificando:

- O nome do CodeCommit repositório em que a ramificação é criada (com a `--repository-name` opção).

 Note

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

- O nome da nova ramificação (com a opção `--branch-name`).
- O ID da confirmação para a qual a nova ramificação apontará (com a opção `--commit-id`).

Por exemplo, para criar uma ramificação chamada `MyNewBranch` que aponte para o ID do commit `317f8570EXAMPLE` em um CodeCommit repositório chamado `MyDemoRepo`:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

Esse comando gerará saídas somente se houver erros.

2. Para atualizar a lista de ramificações de CodeCommit repositório disponíveis em seu repositório local com o novo nome da ramificação remota, execute `git remote update remote-name`

Por exemplo, para atualizar a lista de ramificações disponíveis para o CodeCommit repositório com o apelido `origin`:

```
git remote update origin
```

 Note

Como alternativa, você pode executar o seguinte `git fetch`. Você também pode visualizar todas as ramificações remotas executando `git branch --all`, mas até que você atualize a lista do repositório local, a ramificação remota que você criou não aparecerá nela.

Para obter mais opções, consulte a documentação do Git.

3. Para configurar uma ramificação no repositório local que esteja conectada à nova ramificação no CodeCommit repositório, execute. `git checkout remote-branch-name`

Note

Para obter uma lista dos nomes dos CodeCommit repositórios e seus URLs, execute o `git remote -v` comando.

Limite os envios e fusões às ramificações em AWS CodeCommit

Por padrão, qualquer usuário do CodeCommit repositório que tenha permissões suficientes para enviar código para o repositório pode contribuir com qualquer ramificação desse repositório. Isso é válido independentemente de você adicionar uma ramificação ao repositório usando o console, a linha de comando ou o Git. No entanto, você pode desejar configurar uma ramificação para que apenas alguns usuários do repositório possam enviar por push ou mesclar o código para essa ramificação. Por exemplo, você pode desejar configurar uma ramificação usada para o código de produção para que apenas um subconjunto de desenvolvedores sênior possam enviar por push ou mesclar as alterações para essa ramificação. Outros desenvolvedores ainda podem acionar essa ramificação, fazer suas próprias ramificações e criar solicitações de pull, mas não podem enviar por push ou mesclar as alterações para essa ramificação. Você pode configurar esse acesso criando uma política condicional que usa uma chave de contexto para uma ou mais ramificações no IAM.

Note

Para concluir alguns dos procedimentos deste tópico, você deve fazer login com um usuário administrativo que tenha permissões suficientes para configurar e aplicar as políticas do IAM. Para obter mais informações, consulte [Criação de um usuário administrador e grupo do IAM](#).

Tópicos

- [Configure uma política do IAM para limitar envios por push e mesclagens para ramificações](#)
- [Aplicar a política do IAM a um grupo ou perfil do IAM](#)
- [Teste a política](#)

Configure uma política do IAM para limitar envios por push e mesclagens para ramificações

Você pode criar uma política no IAM que impede os usuários de atualizar uma ramificação, incluindo confirmações de push e mesclagens de solicitações pull para ramificações. Para fazer isso, a sua política usa uma instrução condicional, de modo que o efeito da instrução Deny se aplica somente se a condição for atendida. O APIs que você inclui na Deny declaração determina quais ações não são permitidas. Você pode configurar essa política para aplicar a apenas uma ramificação em um repositório, a uma série de ramificações em um repositório ou a todas as ramificações que correspondem aos critérios em todos os repositórios em uma conta da Amazon Web Services.

Criar uma política condicional para ramificações

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Políticas.
3. Selecione Criar política.
4. Escolha JSON e, em seguida, cole o seguinte exemplo de política. Substitua o valor do Resource pelo ARN do repositório que contém a ramificação à qual você deseja restringir o acesso. Substitua o valor de `codecommit:References` por uma referência à ramificação ou ramificações às quais você deseja restringir o acesso. Por exemplo, essa política nega o envio de confirmações, mesclagem e exclusão de ramificações, exclusão de arquivos, mesclagem de solicitações pull e a inclusão de arquivos a uma ramificação chamada *main* e uma ramificação chamada *prod* em um repositório chamado *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"
        }
    }
}

```

Ramificações no Git são meros ponteiros (referências) ao valor SHA-1 da confirmação head, e é por isso que a condição usa References. A instrução Null é necessária em qualquer política cujo efeito seja Deny e na qual GitPush seja uma das ações. Isso é necessário devido à forma como o Git e git-receive-pack funcionam ao enviar alterações por push de um repositório local ao CodeCommit.

Tip

Para criar uma política que se aplica a todas as ramificações chamadas de principal em todos os repositórios em uma conta da Amazon Web Services, altere o valor de Resource de um ARN de repositório para um asterisco (*).

5. Selecione Revisar política. Corrija todos os erros na sua declaração de política e, em seguida, continue a Create policy (Criar política).
6. Quando o JSON é validado, a página Create policy (Criar política) é exibida. Um aviso é exibido na seção Summary (Resumo), informando que essa política não concede permissões. Isso é esperado.

- Em Name (Nome), insira um nome para essa política, como **DenyChangesToMain**.
- Em Description (Descrição), insira uma descrição da finalidade da política. Isso é opcional, mas recomendado.
- Selecione Criar política.

Aplicar a política do IAM a um grupo ou perfil do IAM

Você criou uma política que limita envios via push e mensagens para ramificações, mas a política não terá efeito até você aplicá-la a um grupo, perfil ou usuário do IAM. Como prática recomendada, considere aplicar a política a um grupo ou perfil do IAM. Aplicar políticas a usuários do IAM individuais não escala bem.

Aplicar a política condicional a um grupo ou função

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, se você quer aplicar a política a um grupo do IAM, selecione Grupos. Se você quer aplicar a política a um perfil que os usuários assumirão, escolha Perfil. Escolha o nome do grupo ou função.
3. Na guia Permissions, escolha Attach Policy.
4. Selecione a política condicional que você criou na lista de políticas e, em seguida, escolha Attach policy (Anexar política).

Para obter mais informações, consulte [Anexar e desanexar políticas do IAM](#).

Teste a política

Você deve testar os efeitos da política que aplicou no grupo ou função para garantir que ela funcione como esperado. Há muitas maneiras de fazer isso. Por exemplo, para testar uma política semelhante à exibida acima, você pode:

- Faça login no CodeCommit console com um usuário do IAM que seja membro de um grupo do IAM que tenha a política aplicada ou assuma uma função que tenha a política aplicada. No console, adicione um arquivo na ramificação em que as restrições se aplicam. Você verá uma mensagem de erro ao tentar salvar um arquivo ou fazer upload dele para essa ramificação. Adicionar um arquivo a uma ramificação diferente. A operação deve ser bem-sucedida.

- Faça login no CodeCommit console com um usuário do IAM que seja membro de um grupo do IAM que tenha a política aplicada ou assuma uma função que tenha a política aplicada. Crie uma solicitação pull que seja mesclada com a ramificação na qual as restrições se aplicam. Você deve ser capaz de criar a solicitação pull, mas obterá um erro se tentar mesclar essa solicitação.
- No terminal ou na linha de comando, crie uma confirmação na ramificação em que as restrições se aplicam e, em seguida, envie essa confirmação para o CodeCommit repositório. Você verá uma mensagem de erro. Confirmações e envios por push feitas de outras ramificações devem funcionar normalmente.

Veja os detalhes da filial em AWS CodeCommit

Você pode usar o CodeCommit console para ver detalhes sobre as ramificações em um CodeCommit repositório. Você pode ver a data da última confirmação para uma ramificação, a mensagem de confirmação e muito mais. Você também pode usar o AWS CLI ou o Git de um repositório local conectado ao repositório. CodeCommit

Tópicos

- [Visualizar detalhes de ramificação \(console\)](#)
- [Visualizar detalhes de ramificação \(Git\)](#)
- [Visualizar detalhes de ramificação \(AWS CLI\)](#)

Visualizar detalhes de ramificação (console)

Use o CodeCommit console para visualizar rapidamente uma lista de ramificações do seu repositório e detalhes sobre as ramificações.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja visualizar detalhes da ramificação.
3. No painel de navegação, escolha Branches.

Developer Tools > CodeCommit > Repositories > MyDemoRepo > Branches

MyDemoRepo

Branches Info Delete branch View branch View last commit Create pull request Create branch

Search:

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-7:00)	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

- O nome da ramificação usada como padrão para o repositório é exibido próximo a Ramificação padrão. Para visualizar detalhes sobre a confirmação mais recente para uma ramificação, escolha a ramificação e, em seguida, escolha View last commit (Visualizar última confirmação). Para visualizar os arquivos e o código em uma ramificação, escolha o nome da ramificação.

Visualizar detalhes de ramificação (Git)

Para usar o Git de um repositório local para ver detalhes sobre as ramificações de rastreamento local e remoto de um CodeCommit repositório, execute o comando. `git branch`

As etapas a seguir foram escritas com a suposição de que você já conectou o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

- Execute o comando `git branch` especificando a opção `--all`.

```
git branch --all
```

- Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
MyNewBranch
* main
remotes/origin/MyNewBranch
```

```
remotes/origin/main
```

O asterisco (*) aparece próximo à ramificação aberta atualmente. As entradas depois disso são referências de rastreamento remoto.

 Tip

git branch mostra ramificações locais.

git branch -r mostra ramificações remotas.

git checkout **existing-branch-name** alterna para o nome da ramificação especificada e, se git branch for executado imediatamente após, exibe-o com um asterisco (*).

git remote update **remote-name** atualiza seu repositório local com a lista de ramificações disponíveis do CodeCommit repositório. (Para obter uma lista dos nomes dos CodeCommit repositórios e seus URLs, execute o git remote -v comando.)

Para obter mais opções, consulte a documentação do Git.

Visualizar detalhes de ramificação (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para visualizar detalhes sobre as ramificações em um CodeCommit repositório, execute um ou mais dos seguintes comandos:

- Para visualizar uma lista de nomes de ramificação, execute [list-branches](#).
- Para visualizar informações sobre uma ramificação específica, execute [get-branch](#).

Como visualizar uma lista de nomes de ramificação

1. Execute o list-branches comando, especificando o nome do CodeCommit repositório (com a -- repository-name opção).

 Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

Por exemplo, para ver detalhes sobre as ramificações em um CodeCommit repositório chamado `MyDemoRepo`:

```
aws codecommit list-branches --repository-name MyDemoRepo
```

2. Se houver êxito, o comando gerará como saída um objeto `branchNameList`, com uma entrada para cada ramificação.

Veja um exemplo de saída com base no comando do exemplo anterior:

```
{
  "branches": [
    "MyNewBranch",
    "main"
  ]
}
```

Como visualizar informações sobre uma ramificação

1. Execute o comando `get-branch`, especificando:
 - O nome do repositório (com a opção `--repository-name`).
 - O nome da ramificação (com a opção `--branch-name`).

Por exemplo, para visualizar informações sobre uma ramificação nomeada `MyNewBranch` em um CodeCommit repositório chamado `MyDemoRepo`:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

2. Se houver êxito, o comando gerará como saída o nome da ramificação e o ID da última confirmação feita a ela.

Veja um exemplo de saída com base no comando do exemplo anterior:

```
{
  "branch": {
    "branchName": "MyNewBranch",
    "commitID": "317f8570EXAMPLE"
  }
}
```

```
}  
}
```

Compare e mescle filiais em AWS CodeCommit

Você pode usar o CodeCommit console para comparar ramificações em um CodeCommit repositório. A comparação de ramificações ajuda você a exibir rapidamente as diferenças entre uma ramificação e a ramificação padrão ou a exibir as diferenças entre as duas ramificações.

Tópicos

- [Comparar uma ramificação com a ramificação padrão](#)
- [Comparar duas ramificações específicas](#)
- [Mesclar duas ramificações \(AWS CLI\)](#)

Comparar uma ramificação com a ramificação padrão

Use o CodeCommit console para visualizar rapidamente as diferenças entre uma ramificação e a ramificação padrão do seu repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja comparar as ramificações.
3. No painel de navegação, selecione Commits (Confirmações) e, em seguida, escolha a guia Compare commits (Comparar confirmações).
4. Em Destination (Destino), escolha o nome da ramificação padrão. Em Source (Origem), escolha a ramificação que deseja comparar à ramificação padrão. Escolha Compare.

Comparar duas ramificações específicas

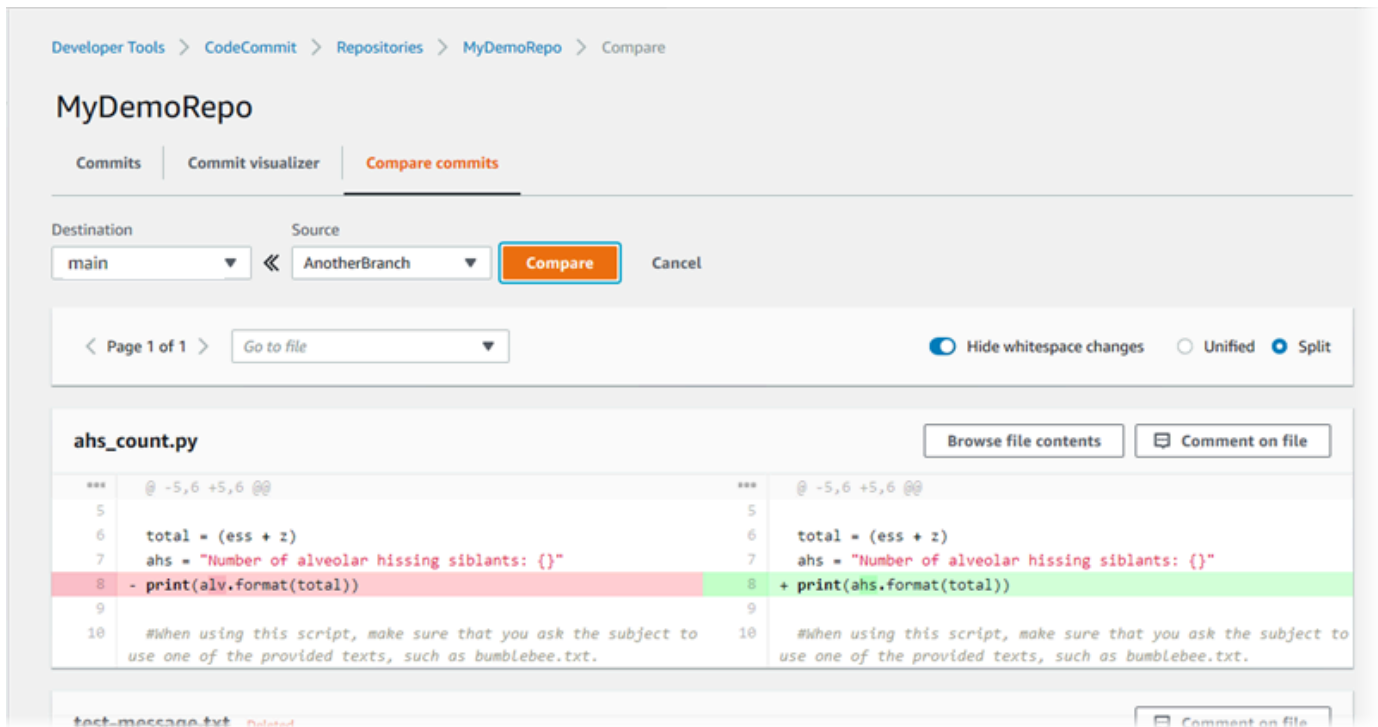
Use o CodeCommit console para ver as diferenças entre duas ramificações que você deseja comparar.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja comparar as ramificações.

3. No painel de navegação, selecione Commits (Confirmações) e, em seguida, escolha a guia Compare commits (Comparar confirmações).
4. Em Destination (Destino) e Source (Origem), escolha as duas ramificações para comparar e depois escolha Compare (Comparar). Para exibir a lista de arquivos alterados, expanda a lista de arquivos alterados. Você pode ver as alterações nos arquivos lado a lado (visualização dividida) ou embutida (exibição unificada).

Note

Se você entrou como um usuário do IAM, é possível configurar e salvar suas preferências para visualizar o código e outras configurações do console. Para obter mais informações, consulte [Trabalhar com preferências do usuário](#).



Mesclar duas ramificações (AWS CLI)

Você pode mesclar duas ramificações em um CodeCommit repositório AWS CLI usando uma das estratégias de mesclagem disponíveis executando um dos seguintes comandos:

- Para mesclar duas ramificações usando a estratégia de mesclagem de avanço rápido, execute o comando [merge-branches-by-fast-forward](#).
- Para mesclar duas ramificações usando a estratégia de mesclagem de compressão, execute o comando [merge-branches-by-squash](#).
- Para mesclar duas ramificações usando a estratégia de mesclagem de três vias, execute o comando [merge-branches-by-three-way](#).

Também é possível testar mesclagens executando o comando `create-unreferenced-merge-commit`. Para obter mais informações, consulte [Resolver conflitos em uma solicitação pull](#).

 Note

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para mesclar duas ramificações em um repositório CodeCommit

1. Para mesclar duas ramificações usando a estratégia de mesclagem de avanço rápido, execute o comando `merge-branches-by-fast-forward`, especificando:
 - O nome da ramificação de origem que contém as alterações que você deseja mesclar (com a opção `--source-commit-specifier`).
 - O nome da ramificação de destino na qual você deseja mesclar as alterações (com a opção `--destination-commit-specifier`).
 - O nome do repositório (com a opção `--repository-name`).

Por exemplo, para mesclar uma ramificação de origem nomeada *bugfix-1234* em uma ramificação de destino nomeada *preprod* em um repositório chamado: *MyDemoRepo*

```
aws codecommit merge-branches-by-fast-forward --source-commit-specifier bugfix-  
bug1234 --destination-commit-specifier preprod --repository-name MyDemoRepo
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{
```

```
"commitId": "4f178133EXAMPLE",  
"treeId": "389765daEXAMPLE"  
}
```

2.

Para mesclar duas ramificações usando a estratégia de mesclagem de compressão, execute o comando `merge-branches-by-squash`, especificando:

- O nome da ramificação de origem que contém as alterações que você deseja mesclar (com a opção `--source-commit-specifier`).
- O nome da ramificação de destino na qual você deseja mesclar as alterações (com a opção `--destination-commit-specifier`).
- O nome do repositório (com a opção `--repository-name`).
- A mensagem de confirmação a ser incluída (com a opção `--commit-message`).
- O nome a ser usado para a confirmação (com a opção `--name`).
- O endereço de e-mail a ser usado para a confirmação (com a opção `--email`).

Por exemplo, para mesclar uma ramificação de origem nomeada *bugfix-bug1234* com uma ramificação de destino nomeada *bugfix-quarterly* em um repositório chamado: *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

3.

Para mesclar duas ramificações usando a estratégia de mesclagem de três vias, execute o comando `merge-branches-by-three-way`, especificando:

- O nome da ramificação de origem que contém as alterações que você deseja mesclar (com a opção `--source-commit-specifier`).

- O nome da ramificação de destino na qual você deseja mesclar as alterações (com a opção `--destination-commit-specifier`).
- O nome do repositório (com a opção `--repository-name`).
- A mensagem de confirmação a ser incluída (com a opção `--commit-message`).
- O nome a ser usado para a confirmação (com a opção `--name`).
- O endereço de e-mail a ser usado para a confirmação (com a opção `--email`).

Por exemplo, para mesclar uma ramificação de origem nomeada *main* com uma ramificação de destino nomeada *bugfix-1234* em um repositório chamado: *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 houver êxito, o comando gerará uma saída semelhante à seguinte:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Alterar as configurações da ramificação em AWS CodeCommit

Você pode alterar qual ramificação usar como ramificação padrão no AWS CodeCommit console ou com AWS CLI o. Por exemplo, se você criou sua primeira confirmação usando um cliente Git que definiu a ramificação padrão como master, você pode criar uma ramificação chamada principal e, em seguida, alterar as configurações da ramificação para que a nova ramificação seja definida como a ramificação padrão do repositório. Para alterar as configurações de outras ramificações, você pode usar o Git de um repositório local conectado ao repositório. CodeCommit

Tópicos

- [Alterar a ramificação padrão \(console\)](#)
- [Alterar a ramificação padrão \(AWS CLI\)](#)

Alterar a ramificação padrão (console)

Você pode especificar qual ramificação é a ramificação padrão em um CodeCommit repositório no AWS CodeCommit console.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja alterar as configurações.
3. No painel de navegação, selecione Configurações.
4. Em Default branch (Ramificação padrão), selecione a lista suspensa de ramificações e escolha uma outra ramificação. Escolha Salvar.

Tip

- Se você não vê outra ramificação na lista suspensa, então você não criou nenhuma ramificação adicional. Você não pode alterar a ramificação padrão de um repositório se o repositório tiver apenas uma ramificação. Para obter mais informações, consulte [Crie uma filial em AWS CodeCommit](#).
- Se você não vê a seção Ramificação padrão, mas vê itens para regras de notificação e conexões, você está no menu de configurações gerais do console. O menu de configurações dos repositórios está listado em Repositórios, no mesmo nível de Código e Solicitações pull.

Alterar a ramificação padrão (AWS CLI)

Para usar AWS CLI comandos com CodeCommit, instale AWS CLI o. Para obter mais informações, consulte [Referência da linha de comando](#).

Para usar o AWS CLI para alterar as configurações de ramificação de um repositório em um CodeCommit repositório, execute o seguinte comando:

- [update-default-branch](#) para alterar a ramificação padrão.

Como alterar a ramificação padrão

1. Execute o comando update-default-branch, especificando:

- O nome do CodeCommit repositório em que a ramificação padrão é atualizada (com a `--repository-name` opção).

 Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

- O nome da nova ramificação padrão (com a opção `--default-branch-name`).

 Tip

Para obter o nome da ramificação, execute o comando [list-branches](#).

2. Por exemplo, para alterar a ramificação padrão para MyNewBranch em um CodeCommit repositório chamado MyDemoRepo:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Esse comando gerará saídas somente se houver erros.

Para obter mais opções, consulte a documentação do Git.

Excluir uma ramificação em AWS CodeCommit

Você pode usar o CodeCommit console para excluir uma ramificação em um repositório. A exclusão de uma ramificação em CodeCommit não exclui essa ramificação em um repositório local, portanto, os usuários podem continuar a ter cópias dessa ramificação até a próxima vez que fizerem as alterações. Para excluir uma ramificação localmente e enviar essa alteração para o CodeCommit repositório, use o Git de um repositório local conectado ao repositório. CodeCommit

A exclusão de uma ramificação não exclui confirmações, mas exclui todas as referências às confirmações nessa ramificação. Se você excluir uma ramificação que contém confirmações que não foram mescladas em outra ramificação no repositório, não poderá recuperar essas confirmações a menos que tenha a confirmação completa. IDs

 Note

Você não pode usar as instruções neste tópico para excluir a ramificação padrão de um repositório. Se você deseja excluir a ramificação padrão, crie uma ramificação, defina a nova ramificação como padrão e, em seguida, exclua a antiga. Para obter mais informações, consulte [Criar uma ramificação](#) e [Alterar configurações de ramificação](#).

Tópicos

- [Excluir uma ramificação \(console\)](#)
- [Excluir uma ramificação \(AWS CLI\)](#)
- [Excluir uma ramificação \(Git\)](#)

Excluir uma ramificação (console)

Você pode usar o CodeCommit console para excluir uma ramificação em um CodeCommit repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositories (Repositórios), escolha o nome do repositório onde você deseja excluir uma ramificação.
3. No painel de navegação, escolha Branches.
4. Encontre o nome da ramificação que você deseja excluir, selecione Delete branch (Excluir ramificação) e confirme sua escolha.

Excluir uma ramificação (AWS CLI)

Você pode usar o AWS CLI para excluir uma ramificação em um CodeCommit repositório, se essa ramificação não for a ramificação padrão do repositório. Para obter mais informações sobre como instalar e usar o AWS CLI, consulte [Referência da linha de comando](#).

1. No terminal ou na linha de comando, execute o comando delete-branch, especificando:
 - O nome do CodeCommit repositório em que a ramificação será excluída (com a --repository-name opção).

 Tip

Para obter o nome do CodeCommit repositório, execute o comando [list-repositories](#).

- O nome da ramificação a ser excluída (com a opção `branch-name`).

 Tip

Para obter o nome da ramificação, execute o comando [list-branches](#).

2. Por exemplo, para excluir uma ramificação nomeada `MyNewBranch` em um CodeCommit repositório chamado `MyDemoRepo`:

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

Esse comando retorna informações sobre a ramificação excluída, incluindo o nome da ramificação excluída e o ID de confirmação completo da confirmação que era o cabeçalho da ramificação. Por exemplo:

```
"deletedBranch": {  
  "branchName": "MyNewBranch",  
  "commitId": "317f8570EXAMPLE"  
}
```

Excluir uma ramificação (Git)

Siga estas etapas para usar o Git de um repositório local para excluir uma ramificação em um repositório. CodeCommit

Essas etapas são escritas com a suposição de que você já conectou o repositório local ao CodeCommit repositório. Para instruções, consulte [Conexão a um repositório](#).

1. Para excluir a ramificação do repositório local, execute o `git branch -D branch-name` comando onde **branch-name** está o nome da ramificação que você deseja excluir.

 Tip

Para obter uma lista de nomes de ramificação, execute `git branch --all`.

Por exemplo, para excluir uma ramificação do repositório local chamada `MyNewBranch`:

```
git branch -D MyNewBranch
```

2. Para excluir a ramificação do CodeCommit repositório, execute o `git push remote-name --delete branch-name` comando em que **remote-name** é o apelido que o repositório local usa para o CodeCommit repositório e **branch-name** é o nome da ramificação que você deseja excluir do repositório. CodeCommit

 Tip

Para obter uma lista dos nomes dos CodeCommit repositórios e seus URLs, execute o `git remote -v` comando.

Por exemplo, para excluir uma ramificação nomeada `MyNewBranch` no CodeCommit repositório chamada `origin`:

```
git push origin --delete MyNewBranch
```

 Tip

Esse comando não excluirá a ramificação se ela for padrão.

Para obter mais opções, consulte a documentação do Git.

Trabalhar com preferências do usuário

Você pode usar o console do AWS CodeCommit para definir algumas das configurações padrão. Por exemplo, você pode alterar suas preferências para visualizar as alterações de código em linha ou em uma visualização dividida. Quando você faz uma alteração em uma dessas configurações, o console do AWS CodeCommit define um cookie em seu navegador que armazena e aplica suas opções sempre que você usa o console. Essas preferências são aplicadas a todos os repositórios em todas as regiões sempre que você acessar o console do AWS CodeCommit usando esse navegador. Essa configuração de preferências não é específica ao repositório ou à região. Elas não têm efeito sobre suas interações com a AWS CLI, a API do AWS CodeCommit ou outros serviços que interagem com o AWS CodeCommit.

Note

Os cookies de preferências de usuário são específicos a um navegador. Se você limpar os cookies de seu navegador, suas preferências serão limpas. Da mesma forma, se você usar um outro navegador para acessar um repositório, o navegador não terá acesso aos cookies do outro navegador. Suas preferências não serão mantidas.

As preferências do usuário incluem:

- Ao visualizar alterações no código, seja para usar a exibição Unified ou Split, e quiser mostrar ou ocultar mudanças no espaço em branco.
- Ao visualizar, editar ou criar código, seja para usar um plano de fundo de luz ou um fundo escuro na janela do editor de código.

Não há uma página para configurar suas preferências. Em vez disso, sempre que você alterar uma preferência no console, como a maneira de visualizar alterações de código, essa alteração será salva e aplicada sempre que for apropriada.

Migração para o AWS CodeCommit

É possível migrar um repositório do Git para um repositório do CodeCommit de várias formas: clonando, espelhando, migrando todas ou apenas algumas ramificações, e assim por diante. Também é possível migrar o conteúdo não versionado local do seu computador para o CodeCommit.

Os seguintes tópicos mostram algumas das possíveis maneiras de migrar um repositório. As etapas podem variar dependendo do tipo, do estilo ou da complexidade do repositório e do que você decidiu sobre o que migrar e como realizar a migração. Para repositórios muito grandes, considere [migrar de maneira incremental](#).

Note

É possível migrar para o CodeCommit a partir de outros sistemas de controle de versão, como Perforce, Subversion ou TFS, mas você deverá migrar antes para o Git. Para obter mais opções, consulte a documentação do Git. Como alternativa, você pode analisar informações sobre [migrar para o Git](#) no livro Pro Git, de Scott Chacon e Ben Straub.

Tópicos

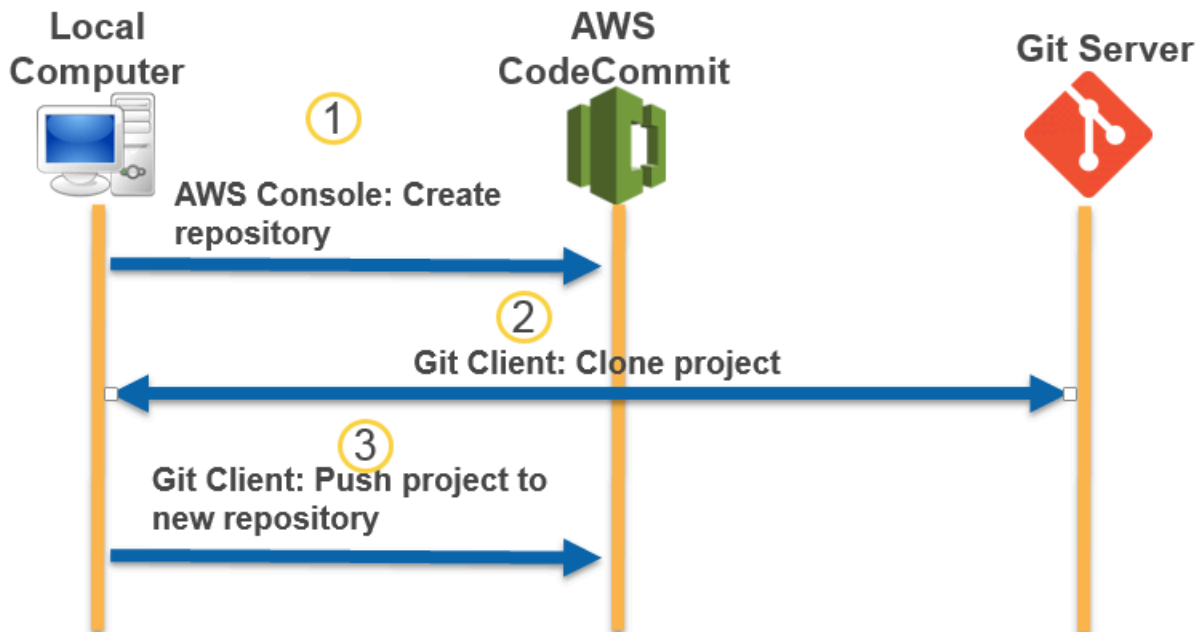
- [Migrar um repositório Git para AWS CodeCommit](#)
- [Migre conteúdo local ou não versionado para AWS CodeCommit](#)
- [Migrar um repositório de forma incremental](#)

Migrar um repositório Git para AWS CodeCommit

Você pode migrar um repositório Git existente para um repositório. CodeCommit Os procedimentos neste tópico mostram como migrar um projeto hospedado em outro repositório Git para o CodeCommit. Como parte do processo, você:

- Conclua a configuração inicial necessária para CodeCommit.
- Crie um CodeCommit repositório.
- Clone o repositório e envie-o para o. CodeCommit
- Visualize arquivos no CodeCommit repositório.

- Compartilhe o CodeCommit repositório com sua equipe.



Tópicos

- [Etapa 0: configuração necessária para acesso ao CodeCommit](#)
- [Etapa 1: criar um CodeCommit repositório](#)
- [Etapa 2: clonar o repositório e enviar para o repositório CodeCommit](#)
- [Etapa 3: Exibir arquivos em CodeCommit](#)
- [Etapa 4: compartilhar o CodeCommit repositório](#)

Etapa 0: configuração necessária para acesso ao CodeCommit

Antes de migrar um repositório para CodeCommit, você deve criar e configurar um usuário do IAM CodeCommit e configurar seu computador local para acesso. Também é necessário instalar o AWS CLI para gerenciar o CodeCommit. Embora você possa realizar a maioria das CodeCommit tarefas sem ele, ele AWS CLI oferece flexibilidade ao trabalhar com o Git na linha de comando ou no terminal.

Se você já está configurado para CodeCommit, pode pular para [Etapa 1: criar um CodeCommit repositório](#).

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório e Autenticação e controle de acesso para AWS CodeCommit](#)

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. `git-remote-codecommit`

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```

Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, certifique-se de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

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

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

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)
- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

- 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

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Em seguida, você deve instalar o Git.

- Para Linux, macOS ou Unix:

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se

Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em [Solução de problemas](#).

- Para Windows:

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como [Git for Windows](#). Se você usar esse link para instalar o Git, poderá aceitar todas as configurações padrão da instalação, exceto as seguintes:

- Quando solicitado durante a etapa Ajustar o seu ambiente PATH, escolha a opção de usar o Git na linha de comando.
- (Opcional) Se você pretende usar HTTPS com o auxiliar de credenciais incluído no, em AWS CLI vez de configurar as credenciais CodeCommit do Git, na página Configurando opções extras, verifique se a opção Habilitar o Git Credential Manager está desmarcada. O Gerenciador de Credenciais do Git só é compatível CodeCommit se os usuários do IAM configurarem as credenciais do Git. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#) e [Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado \(403\)](#).

 Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em [Solução de problemas](#).

CodeCommit suporta autenticação HTTPS e SSH. Para concluir a configuração, você deve configurar as credenciais do Git para CodeCommit (HTTPS, recomendado para a maioria dos usuários), um key pair SSH para usar ao CodeCommit acessar (SSH) git-remote-codecommit (recomendado para usuários que usam acesso federado) ou o auxiliar de credenciais incluído no (HTTPS). AWS CLI

- Para ver as credenciais do Git em todos os sistemas operacionais compatíveis, consulte [Etapa 3: criar credenciais do Git para conexões HTTPS com CodeCommit](#).
- Para SSH no Linux, macOS ou Unix, consulte [SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas para Git e CodeCommit](#).
- Para ver SSH no Windows, consulte [Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit](#).
- Em git-remote-codecommit, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).
- Para o assistente de credenciais no Linux, macOS ou Unix, consulte [Configurar o auxiliar de credenciais \(Linux, macOS ou Unix\)](#).
- Para ver o auxiliar de credenciais no Windows, consulte [Configuração de auxiliar de credenciais \(Windows\)](#).

Etapa 1: criar um CodeCommit repositório

Nesta seção, você usa o CodeCommit console para criar o CodeCommit repositório usado no restante deste tutorial. Para usar o AWS CLI para criar o repositório, consulte [Criar um repositório \(AWS CLI\)](#).

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde você deseja criar o repositório. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), selecione Create repository (Criar repositório).
4. Na página Create repository (Criar repositório), em Repository name (Nome do repositório), insira um nome para o repositório.

Note

Os nomes de repositórios diferenciam maiúsculas de minúsculas. O nome deve ser exclusivo na Região da AWS para sua conta da Amazon Web Services.

5. (Opcional) Em Description (Descrição), insira uma descrição para o repositório. Isso pode ajudar você e outros usuários a identificar a finalidade do repositório.

 Note

O campo de descrição exibe o Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o `GetRepository` ou `BatchGetRepositories` APIs e planeja exibir o campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

6. (Opcional) Escolha Adicionar tag para adicionar uma ou mais tags de repositório (um rótulo de atributo personalizado que ajuda você a organizar e gerenciar seus AWS recursos) ao seu repositório. Para obter mais informações, consulte [Marcando repositórios em AWS CodeCommit](#).
7. (Opcional) Expanda Configuração adicional para especificar se você deseja usar a chave padrão Chave gerenciada pela AWS ou sua própria chave gerenciada pelo cliente para criptografar e descriptografar dados nesse repositório. Se você optar por usar sua própria chave gerenciada pelo cliente, deverá garantir que ela esteja disponível na Região da AWS local em que você está criando o repositório e que a chave esteja ativa. Para obter mais informações, consulte [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#).
8. (Opcional) Selecione Habilitar Amazon CodeGuru Reviewer para Java e Python se esse repositório contiver código Java ou Python e você CodeGuru quiser que o Reviewer o analise. CodeGuru O revisor usa vários modelos de aprendizado de máquina para encontrar defeitos no código e sugerir melhorias e correções nas pull requests. Para obter mais informações, consulte o [Guia do usuário do Amazon CodeGuru Reviewer](#).
9. Escolha Criar.

[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](#)

Após ser criado, o repositório aparecerá na lista Repositories (Repositórios). Na coluna URL, selecione o ícone de cópia e, em seguida, o protocolo (HTTPS ou SSH) a ser usado para se conectar ao CodeCommit. Copie o URL.

Por exemplo, se você nomeou seu repositório *MyClonedRepository* e está usando credenciais do Git com HTTPS na região Leste dos EUA (Ohio), a URL terá a seguinte aparência:

```
https://git-codecommit.us-east-2.amazonaws.com/MyClonedRepository
```

Você precisará desse URL posteriormente em [Etapa 2: clonar o repositório e enviar para o repositório CodeCommit](#).

Etapa 2: clonar o repositório e enviar para o repositório CodeCommit

Nesta seção, você clonará um repositório Git existente para o seu computador local, criando o que é chamado de repositório local. Em seguida, você envia o conteúdo do repositório local para o CodeCommit repositório criado anteriormente.

1. No terminal ou no prompt de comando do seu computador local, execute o git clone comando com a `--mirror` opção de clonar uma cópia vazia do repositório remoto em uma nova pasta chamada. `aws-codecommit-demo` Este é um repositório vazio destinado apenas à migração. Não é o repositório local para interagir com o repositório migrado em. CodeCommit Você pode criar isso mais tarde, após a conclusão da migração para CodeCommit .

O exemplo a seguir clona um aplicativo de demonstração hospedado em GitHub (<https://github.com/aws-labs/aws-demo-php-simple-app.git>) em um repositório local em um diretório chamado. `aws-codecommit-demo`

```
git clone --mirror https://github.com/aws-labs/aws-demo-php-simple-app.git aws-codecommit-demo
```

2. Altere os diretórios para o diretório onde você criou o clone.

```
cd aws-codecommit-demo
```

3. Execute o git push comando, especificando a URL e o nome do CodeCommit repositório de destino e a `--all` opção. (Este é o URL que você copiou em [Etapa 1: criar um CodeCommit repositório](#)).

Por exemplo, se você nomeou seu repositório `MyClonedRepository` e está configurado para usar HTTPS, execute o seguinte comando:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --all
```

Note

A opção `--all` apenas envia por push todas as ramificações para o repositório. Ela não é compatível com o push de outras referências, como tags. Se você deseja enviar tags por push, aguarde até o push inicial ser concluído e, em seguida, envie por push novamente, desta vez usando a opção `--tags`:

```
git push ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyClonedRepository --tags
```

Para obter mais informações, consulte [Git push](#) no site do Git. Para obter informações sobre enviar grandes repositórios por push, principalmente ao enviar todas as

referências por push de uma vez (por exemplo, com a opção `--mirror`), consulte [Migrar um repositório em incrementos](#).

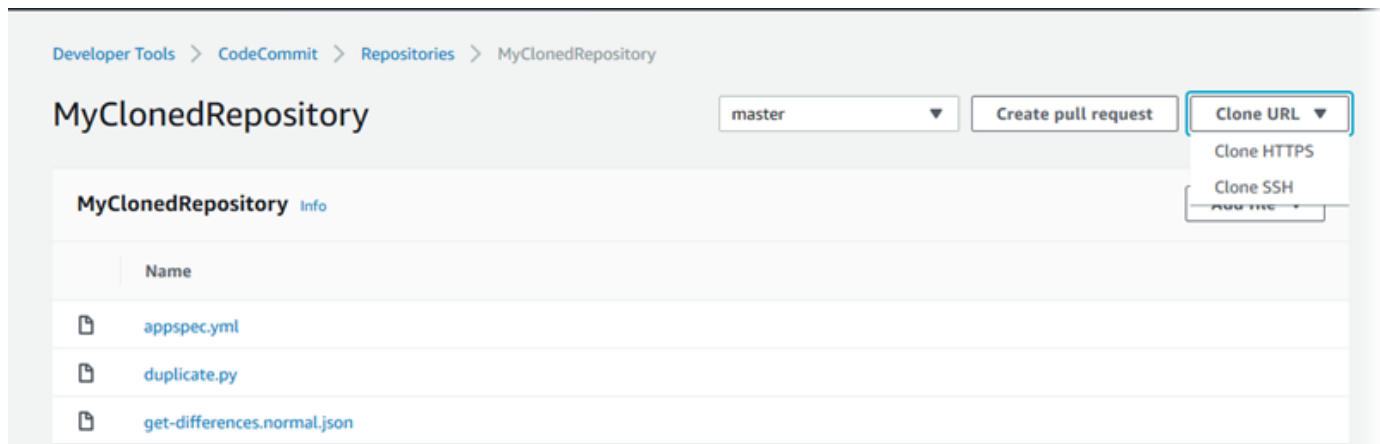
Você pode excluir a `aws-codecommit-demo` pasta e seu conteúdo depois de migrar o repositório para o. CodeCommit Para criar um repositório local com todas as referências corretas para trabalhar com o repositório CodeCommit, execute o `git clone` comando sem a `--mirror` opção:

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

Etapa 3: Exibir arquivos em CodeCommit

Depois de enviar o conteúdo do seu diretório, você pode usar o CodeCommit console para visualizar rapidamente todos os arquivos desse repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositórios, escolha o nome do repositório (por exemplo, `MyClonedRepository`).
3. Visualize os arquivos no repositório das ramificações, do clone URLs, das configurações e muito mais.



Etapa 4: compartilhar o CodeCommit repositório

Quando você cria um repositório no CodeCommit, dois endpoints são gerados: um para conexões HTTPS e outro para conexões SSH. Ambos fornecem conexões seguras em uma rede. Os usuários podem usar um dos dois protocolos. Ambos os endpoints se mantêm ativos, independentemente do protocolo que você recomenda aos usuários. Antes de compartilhar o repositório com outros,

é necessário criar políticas do IAM que permitem o acesso de outros usuários ao seu repositório. Forneça essas instruções de acesso aos seus usuários.

Criação de uma política gerenciada por cliente para seu repositório

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Na área de navegação do Painel, escolha Políticas e, em seguida, Create Policy.
3. Na página Criar políticas, escolha Importar política gerenciada.
4. Na página Importar políticas gerenciadas, em Filtrar políticas, insira o **AWSCodeCommitPowerUser**. Selecione o botão ao lado do nome da política e escolha Importar.
5. Na página Create policy (Criar política), escolha JSON. Substitua a parte “*” da Resource linha para CodeCommit ações pelo Amazon Resource Name (ARN) do CodeCommit repositório, conforme mostrado aqui:

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

 Tip

Para encontrar o ARN do CodeCommit repositório, acesse o CodeCommit console, escolha o nome do repositório na lista e escolha Configurações. Para obter mais informações, consulte [Visualizar detalhes de repositórios](#).

Se você quiser que essa política seja aplicada a mais de um repositório, adicione cada um como um recurso especificando o ARN dele. Inclua uma vírgula entre cada instrução de recurso, conforme mostrado aqui:

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

Ao concluir a edição, selecione Revisar política.

6. Na página Revisar Política, em Nome, insira um novo nome para a política (por exemplo, *AWSCodeCommitPowerUser-MyDemoRepo*). Como opção, você pode fornecer uma descrição para essa política.
7. Escolha Create Policy.

Para gerenciar o acesso ao repositório, crie um grupo do IAM para os usuários, adicione os usuários do IAM ao grupo e anexe a política gerenciada pelo cliente que você criou na etapa anterior. Anexe quaisquer outras políticas necessárias para acesso, como IAMUser SSHKeys ou IAMSelfManageServiceSpecificCredentials.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Na área de navegação Painel, escolha Grupos e Criar novo grupo.
3. Na página Set Group Name (Definir nome de grupo), em Group Name (Nome do grupo), insira um nome para o grupo (por exemplo, *MyDemoRepoGroup*) e escolha Next Step (Próxima etapa). Considere incluir o nome do repositório como parte do nome do grupo.

 Note

Esse nome precisa ser exclusivo em toda a conta da Amazon Web Services.

4. Selecione a caixa ao lado da política gerenciada pelo cliente que você criou na seção anterior (por exemplo, *AWSCodeCommitPowerUser-MyDemoRepo*).
5. Na página Análise, escolha Criar grupo. O IAM criará esse grupo com as políticas especificadas já anexadas. O grupo aparecerá na lista de grupos associados à sua conta da Amazon Web Services.
6. Escolha seu grupo na lista.
7. Na página de resumo do grupo, escolha a guia Usuários e Adicionar usuários ao grupo. Na lista que mostra todos os usuários associados à sua conta da Amazon Web Services, selecione as caixas ao lado dos usuários aos quais você deseja permitir acesso ao CodeCommit repositório e escolha Adicionar usuários.

 Tip

Você pode usar a caixa Pesquisar para encontrar rapidamente usuários por nome.

8. Depois de adicionar os usuários, feche o console do IAM.

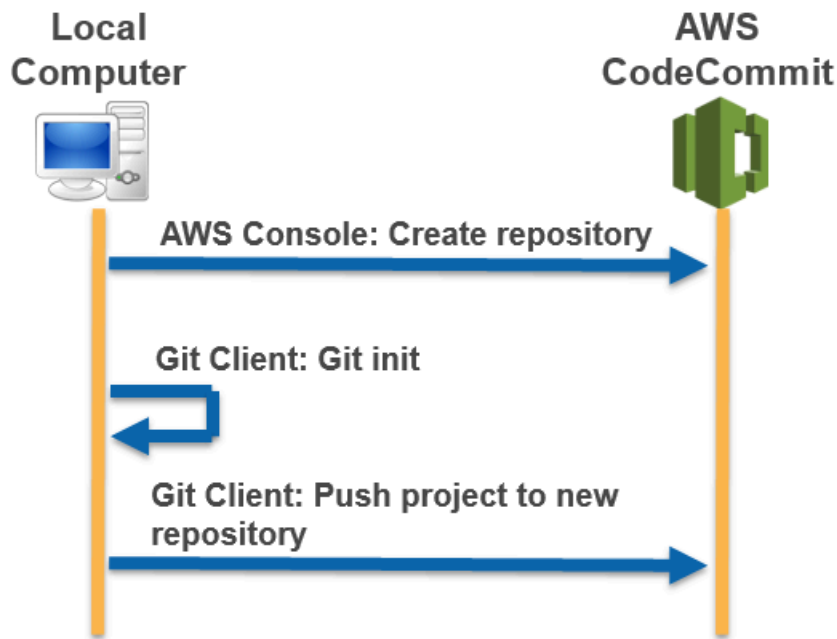
Depois de criar um usuário do IAM para acessar CodeCommit usando o grupo de políticas e as políticas que você configurou, envie a esse usuário as informações necessárias para se conectar ao repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), escolha o repositório que deseja atualizar.
4. Em Clone URL (Clonar URL), escolha o protocolo que você deseja que os usuários usem. Isso copia a URL clonada para o protocolo de conexão.
5. Envie aos usuários o URL do clone junto com quaisquer outras instruções, como instalar o AWS CLI, configurar um perfil ou instalar o Git. Não se esqueça de incluir as informações de configuração para o protocolo de conexão (por exemplo, HTTPS).

Migre conteúdo local ou não versionado para AWS CodeCommit

Os procedimentos deste tópico mostram como migrar um projeto existente ou conteúdo local em seu computador para um CodeCommit repositório. Como parte do processo, você:

- Conclua a configuração inicial necessária para CodeCommit.
- Crie um CodeCommit repositório.
- Coloque uma pasta local no controle de versão do Git e envie o conteúdo dessa pasta para o CodeCommit repositório.
- Visualize arquivos no CodeCommit repositório.
- Compartilhe o CodeCommit repositório com sua equipe.



Tópicos

- [Etapa 0: configuração necessária para acessar a CodeCommit](#)
- [Etapa 1: criar um CodeCommit repositório](#)
- [Etapa 2: migrar o conteúdo local para o repositório CodeCommit](#)
- [Etapa 3: Exibir arquivos em CodeCommit](#)
- [Etapa 4: compartilhar o CodeCommit repositório](#)

Etapa 0: configuração necessária para acessar a CodeCommit

Antes de migrar o conteúdo local para CodeCommit, você deve criar e configurar um usuário do IAM CodeCommit e configurar seu computador local para acesso. Também é necessário instalar o AWS CLI para gerenciar o CodeCommit. Embora você possa realizar a maioria das CodeCommit tarefas sem ele, ele AWS CLI oferece flexibilidade ao trabalhar com o Git.

Se você já está configurado para CodeCommit, pode pular para [Etapa 1: criar um CodeCommit repositório](#).

Para criar e configurar um usuário do IAM para acessar CodeCommit

1. Crie uma conta da Amazon Web Services acessando <http://aws.amazon.com> e escolhendo Cadastrar-se.
2. Crie um usuário do IAM ou use um existente na sua conta da Amazon Web Services. Confirme se você tem um ID de chave de acesso e uma chave de acesso secreta associados ao usuário do IAM. Para obter mais informações, consulte [Criar um usuário do IAM na conta da Amazon Web Services](#).

 Note

CodeCommit requer AWS Key Management Service. Se você estiver usando um usuário do IAM existente, certifique-se de que não haja políticas anexadas ao usuário que neguem expressamente as AWS KMS ações exigidas pelo CodeCommit. Para obter mais informações, consulte [AWS KMS e criptografia](#).

3. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
4. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha o usuário do IAM que você deseja configurar para CodeCommit acesso.
5. Na guia Permissions, escolha Add Permissions.
6. Em Grant permissions, escolha Attach existing policies directly.
7. Na lista de políticas, selecione AWSCodeCommitPowerUser ou outra política gerenciada para CodeCommit acesso. Para obter mais informações, consulte [AWS políticas gerenciadas para CodeCommit](#).

Depois de selecionar a política que deseja anexar, escolha Próximo: Revisar para revisar a lista de políticas a serem anexadas ao usuário do IAM. Se a lista estiver correta, selecione Add permissions.

Para obter mais informações sobre políticas CodeCommit gerenciadas e compartilhamento de acesso a repositórios com outros grupos e usuários, consulte [Compartilhar um repositório](#) e [Autenticação e controle de acesso para AWS CodeCommit](#)

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. `git-remote-codecommit`

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```

Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, certifique-se de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

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

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

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)
- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

- 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

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Em seguida, você deve instalar o Git.

- Para Linux, macOS ou Unix:

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como o [Git Downloads](#).

Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se

Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em: [Solução de problemas](#)

- Para Windows:

Para trabalhar com arquivos, commits e outras informações em CodeCommit repositórios, você deve instalar o Git na sua máquina local. CodeCommit suporta as versões 1.7.9 e posteriores do Git. A versão 2.28 do Git é compatível com a configuração do nome de ramificações para as confirmações iniciais. Recomendamos o uso de uma versão recente do Git.

Para instalar o Git, recomendamos sites como [Git for Windows](#). Se você usar esse link para instalar o Git, poderá aceitar todas as configurações padrão da instalação, exceto as seguintes:

- Quando solicitado durante a etapa Ajustar o seu ambiente PATH, escolha a opção de usar o Git na linha de comando.
- (Opcional) Se você pretende usar HTTPS com o auxiliar de credenciais incluído no, em AWS CLI vez de configurar as credenciais CodeCommit do Git, na página Configurando opções extras, verifique se a opção Habilitar o Git Credential Manager está desmarcada. O Gerenciador de Credenciais do Git só é compatível CodeCommit se os usuários do IAM configurarem as credenciais do Git. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#) e [Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado \(403\)](#).

 Note

O Git é uma plataforma em constante evolução atualizada regularmente. Ocasionalmente, uma alteração de recurso pode afetar a forma como funciona com CodeCommit. Se você encontrar problemas com uma versão específica do Git and CodeCommit, revise as informações em: [Solução de problemas](#)

CodeCommit suporta autenticação HTTPS e SSH. Para concluir a configuração, você deve configurar as credenciais do Git para CodeCommit (HTTPS, recomendado para a maioria dos usuários), um key pair SSH (SSH) para usar ao CodeCommit acessar git-remote-codecommit (recomendado para usuários que usam acesso federado) ou o auxiliar de credenciais incluído no AWS CLI

- Para ver as credenciais do Git em todos os sistemas operacionais compatíveis, consulte [Etapa 3: criar credenciais do Git para conexões HTTPS com CodeCommit](#).
- Para SSH no Linux, macOS ou Unix, consulte [SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas para Git e CodeCommit](#).
- Para ver SSH no Windows, consulte [Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit](#).
- Em git-remote-codecommit, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).
- Para o assistente de credenciais no Linux, macOS ou Unix, consulte [Configurar o auxiliar de credenciais \(Linux, macOS ou Unix\)](#).
- Para ver o auxiliar de credenciais no Windows, consulte [Configuração de auxiliar de credenciais \(Windows\)](#).

Etapa 1: criar um CodeCommit repositório

Nesta seção, você usa o CodeCommit console para criar o CodeCommit repositório usado no restante deste tutorial. Para usar o AWS CLI para criar o repositório, consulte [Criar um repositório \(AWS CLI\)](#).

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. No seletor de região, escolha Região da AWS onde você deseja criar o repositório. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), selecione Create repository (Criar repositório).
4. Na página Create repository (Criar repositório), em Repository name (Nome do repositório), insira um nome para o repositório.

Note

Os nomes de repositórios diferenciam maiúsculas de minúsculas. O nome deve ser exclusivo na Região da AWS para sua conta da Amazon Web Services.

5. (Opcional) Em Description (Descrição), insira uma descrição para o repositório. Isso pode ajudar você e outros usuários a identificar a finalidade do repositório.

 Note

O campo de descrição exibe o Markdown no console e aceita todos os caracteres HTML e caracteres Unicode válidos. Se você é um desenvolvedor de aplicativos que está usando o `GetRepository` ou `BatchGetRepositories` APIs e planeja exibir o campo de descrição do repositório em um navegador da Web, consulte a [Referência da CodeCommit API](#).

6. (Opcional) Escolha Adicionar tag para adicionar uma ou mais tags de repositório (um rótulo de atributo personalizado que ajuda você a organizar e gerenciar seus AWS recursos) ao seu repositório. Para obter mais informações, consulte [Marcando repositórios em AWS CodeCommit](#).
7. (Opcional) Expanda Configuração adicional para especificar se você deseja usar a chave padrão Chave gerenciada pela AWS ou sua própria chave gerenciada pelo cliente para criptografar e descriptografar dados nesse repositório. Se você optar por usar sua própria chave gerenciada pelo cliente, deverá garantir que ela esteja disponível no Região da AWS local em que você está criando o repositório e que a chave esteja ativa. Para obter mais informações, consulte [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#).
8. (Opcional) Selecione Habilitar Amazon CodeGuru Reviewer para Java e Python se esse repositório contiver código Java ou Python e você CodeGuru quiser que o Reviewer o analise. CodeGuru O revisor usa vários modelos de aprendizado de máquina para encontrar defeitos no código e sugerir melhorias e correções nas pull requests. Para obter mais informações, consulte o [Guia do usuário do Amazon CodeGuru Reviewer](#).
9. Escolha Criar.

Após ser criado, o repositório aparecerá na lista Repositories (Repositórios). Na coluna URL, selecione o ícone de cópia e, em seguida, o protocolo (HTTPS ou SSH) a ser usado para se conectar ao CodeCommit. Copie o URL.

Por exemplo, se você nomeou seu repositório *MyFirstRepo* e está usando HTTPS, a URL teria a seguinte aparência:

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

Você precisará desse URL posteriormente em [Etapa 2: migrar o conteúdo local para o repositório CodeCommit](#).

Etapa 2: migrar o conteúdo local para o repositório CodeCommit

Agora que você tem um CodeCommit repositório, pode escolher um diretório no seu computador local para convertê-lo em um repositório Git local. O comando `git init` pode ser usado para converter conteúdo existente e não versionado para um repositório do Git ou, se você ainda não tiver arquivos ou conteúdo, para inicializar um repositório novo e vazio.

1. No terminal ou na linha de comando do seu computador local, altere diretórios para o diretório que deseja usar como origem do seu repositório.
2. Execute o comando a seguir para configurar o Git para usar uma ramificação padrão denominada **main**:

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

Você também pode executar esse comando para definir o nome da ramificação padrão como **main** para todos os repositórios recém-criados:

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

3. Execute o comando `git init` para inicializar o controle de versão do Git no diretório. Isso cria um subdiretório `.git` na raiz do diretório que permite o rastreamento do controle de versão. A pasta `.git` também contém todos os metadados necessários para o repositório.

```
git init
```

4. Verifique o status do diretório inicializado executando o seguinte comando:

```
git status
```

Adicione os arquivos que deseja ao controle de versão. Neste tutorial, você executará o comando `git add` com o especificador `.` para adicionar todos os arquivos no diretório. Para outras opções, consulte a documentação do Git.

```
git add .
```

5. Crie uma confirmação para os arquivos adicionados com uma mensagem de confirmação.

```
git commit -m "Initial commit"
```

6. Execute o git push comando, especificando a URL e o nome do CodeCommit repositório de destino e a --all opção. (Este é o URL que você copiou em [Etapa 1: criar um CodeCommit repositório](#).)

Por exemplo, se você nomeou seu repositório *MyFirstRepo* e está configurado para usar HTTPS, execute o seguinte comando:

```
git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyFirstRepo --all
```

Etapa 3: Exibir arquivos em CodeCommit

Depois de enviar o conteúdo do seu diretório, você pode usar o CodeCommit console para visualizar rapidamente todos os arquivos no repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. Em Repositórios, escolha o nome do repositório (por exemplo, *MyFirstRepository*) na lista.
3. Visualize os arquivos no repositório para ramificações, clones URLs, configurações e muito mais.

Etapa 4: compartilhar o CodeCommit repositório

Quando você cria um repositório no CodeCommit, dois endpoints são gerados: um para conexões HTTPS e outro para conexões SSH. Ambos fornecem conexões seguras em uma rede. Os usuários podem usar um dos dois protocolos. Ambos os endpoints se mantêm ativos, independentemente do protocolo que você recomenda aos usuários. Antes de compartilhar o repositório com outros, é necessário criar políticas do IAM que permitem o acesso de outros usuários ao seu repositório. Forneça essas instruções de acesso aos seus usuários.

Criação de uma política gerenciada por cliente para seu repositório

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Na área de navegação do Painel, escolha Políticas e, em seguida, Create Policy.
3. Na página Criar políticas, escolha Importar política gerenciada.

4. Na página Importar políticas gerenciadas, em Filtrar políticas, insira o **AWSCodeCommitPowerUser**. Selecione o botão ao lado do nome da política e escolha Importar.
5. Na página Create policy (Criar política), escolha JSON. Substitua a parte “*” da Resource linha para CodeCommit ações pelo Amazon Resource Name (ARN) do CodeCommit repositório, conforme mostrado aqui:

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

 Tip

Para encontrar o ARN do CodeCommit repositório, acesse o CodeCommit console, escolha o nome do repositório na lista e escolha Configurações. Para obter mais informações, consulte [Visualizar detalhes de repositórios](#).

Se você quiser que essa política seja aplicada a mais de um repositório, adicione cada um como um recurso especificando o ARN dele. Inclua uma vírgula entre cada instrução de recurso, conforme mostrado aqui:

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

Ao concluir a edição, selecione Revisar política.

6. Na página Revisar Política, em Nome, insira um novo nome para a política (por exemplo, **AWSCodeCommitPowerUser-MyDemoRepo**). Como opção, você pode fornecer uma descrição para essa política.
7. Escolha Create Policy.

Para gerenciar o acesso ao repositório, crie um grupo do IAM para os usuários, adicione os usuários do IAM ao grupo e anexe a política gerenciada pelo cliente que você criou

na etapa anterior. Anexe quaisquer outras políticas necessárias para o acesso, como `IAMSelfManageServiceSpecificCredentials` ou `IAMUserSSHKeys`.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Na área de navegação Painel, escolha Grupos e Criar novo grupo.
3. Na página Set Group Name (Definir nome de grupo), em Group Name (Nome do grupo), insira um nome para o grupo (por exemplo, *MyDemoRepoGroup*) e escolha Next Step (Próxima etapa). Considere incluir o nome do repositório como parte do nome do grupo.

 Note

Esse nome precisa ser exclusivo em toda a conta da Amazon Web Services.

4. Selecione a caixa ao lado da política gerenciada pelo cliente que você criou na seção anterior (por exemplo, `AWSCodeCommitPowerUser-MyDemoRepo`).
5. Na página Análise, escolha Criar grupo. O IAM criará esse grupo com as políticas especificadas já anexadas. O grupo aparecerá na lista de grupos associados à sua conta da Amazon Web Services.
6. Escolha seu grupo na lista.
7. Na página de resumo do grupo, escolha a guia Usuários e Adicionar usuários ao grupo. Na lista que mostra todos os usuários associados à sua conta da Amazon Web Services, selecione as caixas ao lado dos usuários aos quais você deseja permitir acesso ao CodeCommit repositório e escolha Adicionar usuários.

 Tip

Você pode usar a caixa Pesquisar para encontrar rapidamente usuários por nome.

8. Depois de adicionar os usuários, feche o console do IAM.

Depois de criar um usuário do IAM para ser usado para acessar CodeCommit usando o grupo de políticas e as políticas que você configurou, envie a esse usuário as informações necessárias para se conectar ao repositório.

1. Abra o CodeCommit console em <https://console.aws.amazon.com/codesuite/codecommit/home>.

2. No seletor de região, escolha Região da AWS onde o repositório foi criado. Os repositórios são específicos para um Região da AWS. Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).
3. Na página Repositories (Repositórios), escolha o repositório que deseja atualizar.
4. Em Clone URL (Clonar URL), escolha o protocolo que você deseja que os usuários usem. Isso copia a URL clonada para o protocolo de conexão.
5. Envie aos usuários o URL do clone junto com quaisquer outras instruções, como instalar o AWS CLI, configurar um perfil ou instalar o Git. Não se esqueça de incluir as informações de configuração para o protocolo de conexão (por exemplo, HTTPS).

Migrar um repositório de forma incremental

Ao migrar para AWS CodeCommit, considere enviar seu repositório em incrementos ou partes para reduzir as chances de um problema de rede intermitente ou um desempenho de rede degradado fazer com que todo o push falhe. Ao usar pushes em incrementos com um script como o que está incluído aqui, você poderá reiniciar a migração e enviar por push somente as confirmações que não obtiveram êxito na tentativa anterior.

Os procedimentos neste tópico mostram como criar e executar um script que migra o repositório em incrementos e reenvia por push somente os incrementos que não obtiveram êxito até a migração estar concluída.

Essas instruções foram escritas presumindo que você já tenha concluído as etapas em [Configurando e Criar um repositório](#).

Tópicos

- [Etapa 0: determinar se a migração será incremental](#)
- [Etapa 1: instalar os pré-requisitos e adicionar o CodeCommit repositório como remoto](#)
- [Etapa 2: criar o script a ser usado na migração em incrementos](#)
- [Etapa 3: Executar o script e migrar incrementalmente para CodeCommit](#)
- [Apêndice: Amostra de script incremental-repo-migration.py](#)

Etapa 0: determinar se a migração será incremental

Há vários fatores a considerar para determinar o tamanho total do seu repositório e se migrará de maneira incremental. O mais óbvio é o tamanho total dos artefatos no repositório. Fatores como

histórico acumulado do repositório também podem contribuir para o tamanho. Um repositório com anos de histórico e ramificações pode ser muito grande, mesmo que os recursos individuais não sejam. Há uma série de estratégias que você pode utilizar para tornar a migração desses repositórios mais simples e mais eficiente. Por exemplo, você pode usar uma estratégia de clonagem superficial ao clonar um repositório com um longo histórico de desenvolvimento, ou pode desativar a compactação delta para grandes arquivos binários. Você pode pesquisar opções consultando a documentação do Git ou optar pela configuração de push em incrementos para migrar o seu repositório usando a amostra de script incluída neste tópico, `incremental-repo-migration.py`

Você pode configurar pushes em incrementos se uma ou mais das seguintes condições forem verdadeiras:

- O repositório que você quer migrar tem um histórico de mais de cinco anos.
- Sua conexão com a Internet está sujeita a interrupções intermitentes, queda de pacotes, resposta lenta ou outras interrupções no serviço.
- O tamanho total do repositório é maior que 2 GB e você pretende migrá-lo por inteiro.
- O repositório contém grandes artefatos ou binários que não são bem compactados, como arquivos grandes de imagem de mais de cinco versões controladas.
- Você já tentou migrar CodeCommit e recebeu uma mensagem de “Erro de serviço interno”.

Mesmo se nenhuma das condições acima for verdadeira, você ainda poderá escolher enviar pushes de maneira incremental.

Etapa 1: instalar os pré-requisitos e adicionar o CodeCommit repositório como remoto

É possível criar o seu próprio script personalizado que tenha pré-requisitos próprios. Se você usar os exemplos incluídos neste tópico, deverá:

- Instalar seus pré-requisitos.
- Clonar o repositório para o computador local.
- Adicione o CodeCommit repositório como remoto para o repositório que você deseja migrar.

Configure para executar incremental-repo-migration o.py

1. No computador local, instale o Python 2.6 ou posterior. Para obter mais informações sobre as versões mais recentes, consulte [o site da Python](#).
2. No mesmo computador, instale GitPython, que é uma biblioteca Python usada para interagir com repositórios Git. Para obter mais informações, consulte a [a documentação do GitPython](#).
3. Use o comando `git clone --mirror` para clonar o repositório que você deseja migrar para o seu computador local. No terminal (Linux, macOS ou Unix) ou no prompt de comando (Windows), use o comando `git clone --mirror` para criar um repositório local para o repositório, incluindo o diretório onde você deseja criá-lo. Por exemplo, para clonar um repositório Git *MyMigrationRepo* chamado com uma URL `https://example.com/my-repo/` de para um diretório chamado: *my-repo*

```
git clone --mirror https://example.com/my-repo/MyMigrationRepo.git my-repo
```

Você verá uma saída semelhante à seguinte, que indica que o repositório foi clonado em um repositório local bare chamado *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. Altere os diretórios para o repositório local do repositório que você acabou de clonar (por exemplo,). *my-repo* Desse diretório, use o comando `git remote add` **DefaultRemoteName RemoteRepositoryURL** para adicionar o repositório CodeCommit como um repositório remoto para o repositório local.

Note

Ao enviar grandes repositórios por push, considere utilizar SSH em vez de HTTPS. Ao enviar por push uma grande alteração, um grande número de alterações ou um grande repositório, as conexões HTTPS de execução prolongada costumam ser encerradas prematuramente devido a problemas de rede ou configurações do firewall. Para obter

mais informações sobre a configuração CodeCommit do SSH, consulte [Para conexões SSH no Linux, macOS ou Unix](#) ou [Para conexões SSH no Windows](#).

Por exemplo, use o comando a seguir para adicionar o endpoint SSH para um CodeCommit repositório chamado MyDestinationRepo como repositório remoto para o remoto chamado: codecommit

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

 Tip

Por ser um clone, o nome do repositório remoto padrão (origin) já está em uso. Você deverá usar outro nome de repositório remoto. Embora o exemplo use codecommit, você pode usar o nome que quiser. Use o comando git remote show para analisar a lista de conjunto de repositórios remotos do seu repositório local.

5. Use o comando git remote -v para exibir as configurações de busca e push do seu repositório local e confirmar se estão corretas. Por exemplo:

```
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 você ainda vir entradas de busca e push de um repositório remoto diferente (por exemplo, entradas para origin), use o comando git remote set-url --delete para removê-las.

Etapa 2: criar o script a ser usado na migração em incrementos

Estas etapas foram escritas presumindo que você esteja usando o script de exemplo incremental-repo-migration.py.

1. Abra um editor de texto e cole os conteúdos da [amostra de script](#) em um documento vazio.
2. Salve o documento em um diretório de documentos (não o diretório de trabalho do seu repositório local) e nomeie-o `incremental-repo-migration.py`. Verifique se o diretório escolhido está configurado no ambiente local ou nas variáveis PATH, para que seja possível executar o script Python a partir de uma linha de comando ou um terminal.

Etapa 3: Executar o script e migrar incrementalmente para CodeCommit

Agora que você criou seu `incremental-repo-migration.py` script, você pode usá-lo para migrar incrementalmente um repositório local para um repositório. CodeCommit Por padrão, o script envia confirmações por push em lotes de 1.000 confirmações e tenta usar as configurações do Git para o diretório de onde é executado como configurações para os repositórios local e remoto. Você pode usar as opções incluídas em `incremental-repo-migration.py` para definir outras configurações, se necessário.

1. No terminal ou na linha de comando, altere os diretórios para o repositório local que você deseja migrar.
2. Nesse diretório, execute o seguinte comando:

```
python incremental-repo-migration.py
```

3. O script executa e mostra progresso no terminal ou na linha de comando. Alguns repositórios grandes demoram para mostrar o progresso. O script será interrompido se um único push falhar três vezes. Então, você poderá executar novamente o script, que começará pelo lote que falhou. Execute o script novamente até que todos os pushes obtenham êxito e a migração seja concluída.

Tip

Você pode executar `incremental-repo-migration.py` a partir de qualquer diretório desde que utilize as opções `-l` e `-r` para especificar as configurações locais e remotas a serem usadas. Por exemplo, para usar o script de qualquer diretório para migrar um repositório local localizado em `/tmp/my-repo` para um remoto apelidado: `codecommit`

```
python incremental-repo-migration.py -l "/tmp/my-repo" -r "codecommit"
```

Você também pode querer usar a opção `-b` para alterar o tamanho do lote padrão usado ao enviar por push de forma incremental. Por exemplo, se você estiver enviando por push regularmente um repositório de arquivos binários muito grandes que costumam mudar frequentemente e trabalham de um local com largura de banda da rede restrita, use a opção `-b` para alterar o tamanho do lote para 500 em vez de 1.000. Por exemplo:

```
python incremental-repo-migration.py -b 500
```

Isso envia o repositório local por push de maneira incremental em lotes de 500 confirmações. Se você decidir alterar o tamanho do lote novamente ao migrar o repositório (por exemplo, se você decidir reduzir o tamanho do lote após uma tentativa falha), lembre-se de usar a opção `-c` para remover as tags do lote antes de redefinir o tamanho do lote com `-b`:

```
python incremental-repo-migration.py -c
python incremental-repo-migration.py -b 250
```

Important

Não use a opção `-c` se desejar executar novamente o script após uma falha. A opção `-c` elimina as tags usadas para acomodar as confirmações em lote. Use a opção `-c` somente se desejar alterar o tamanho do lote e iniciar novamente ou se decidir que não quer mais usar o script.

Apêndice: Amostra de script **incremental-repo-migration.py**

Para a sua conveniência, desenvolvemos um amostra de script Python, `incremental-repo-migration.py`, para enviar um repositório por push de maneira incremental. Esse script é um exemplo de código aberto e é fornecido como se encontra.

```
# 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
)
```


Segurança no AWS CodeCommit

A segurança na nuvem da AWS é nossa maior prioridade. Como cliente da AWS, você se beneficiará de um data center e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem:** a AWS é responsável pela proteção da infraestrutura que executa serviços AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores terceirizados testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao AWS CodeCommit, consulte [Serviços da AWS em escopo por programa de conformidade](#).
- **Segurança na nuvem:** sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o CodeCommit. Os tópicos a seguir mostram como configurar o CodeCommit para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros produtos da AWS que ajudam a monitorar e proteger os recursos do CodeCommit.

Tópicos

- [Proteção de dados no AWS CodeCommit](#)
- [Identity and Access Management para AWS CodeCommit](#)
- [Resiliência no AWS CodeCommit](#)
- [Segurança da infraestrutura no AWS CodeCommit](#)

Proteção de dados no AWS CodeCommit

Por ser um serviço gerenciado, é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte [Segurança na Nuvem AWS](#). Para projetar seu ambiente da AWS usando as práticas recomendadas

de segurança da infraestrutura, consulte [Proteção de Infraestrutura](#) em Pilar de Segurança: AWS Well-Architected Framework.

Você usa chamadas de API publicadas pela AWS para acessar por meio da rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Os repositórios do CodeCommit são criptografados automaticamente quando estão em repouso. Não será necessária nenhuma ação do cliente. O CodeCommit também criptografa dados em trânsito do repositório. Também é possível usar o protocolo HTTPS, o protocolo SSH ou ambos com repositórios do CodeCommit. Para obter mais informações, consulte [Configuração do AWS CodeCommit](#). Também é possível configurar o [acesso entre contas](#) para repositórios do CodeCommit.

Tópicos

- [AWS Key Management Service e criptografia para AWS CodeCommit repositórios](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

AWS Key Management Service e criptografia para AWS CodeCommit repositórios

Os dados nos CodeCommit repositórios são criptografados em trânsito e em repouso. Quando os dados são enviados para um CodeCommit repositório (por exemplo, por meio de uma chamadagit push), CodeCommit criptografa os dados recebidos à medida que são armazenados no repositório. Quando os dados são extraídos de um CodeCommit repositório (por exemplo, por meio de uma chamadagit pull), os CodeCommit descriptografa e os envia para o chamador. Isso pressupõe que o usuário do IAM associado à solicitação push ou pull tenha sido autenticado por. AWS Os dados enviados ou recebidos são transferidos usando protocolos de rede criptografada HTTPS ou SSH.

Você pode usar uma chave gerenciada pelo cliente Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente para criptografar e descriptografar os dados no seu repositório. Consulte mais informações sobre as diferenças entre chaves gerenciadas pelo cliente e Chaves gerenciadas pela AWS em [Customer managed keys and Chaves gerenciadas pela AWS](#). Se você não especificar

uma chave gerenciada pelo cliente, CodeCommit usará uma Chave gerenciada pela AWS para criptografar e descriptografar os dados no seu repositório. Essa Chave gerenciada pela AWS é criada automaticamente para você em sua Conta da AWS. Na primeira vez que você cria um CodeCommit repositório em um novo Região da AWS em sua conta da Amazon Web Services, se você não especificar uma chave gerenciada pelo cliente, CodeCommit cria uma Chave gerenciada pela AWS (a `aws/codecommit` chave) na mesma Região da AWS em AWS Key Management Service (AWS KMS). Essa `aws/codecommit` chave é usada somente por CodeCommit. Ela é armazenada na sua conta da Amazon Web Services. Dependendo do que você especificar, CodeCommit use a chave gerenciada pelo cliente ou a Chave gerenciada pela AWS para criptografar e descriptografar os dados no repositório.

 Important

CodeCommit executa as seguintes AWS KMS ações em relação à AWS KMS chave usada para criptografar e descriptografar dados em um repositório. Se você estiver usando um Chave gerenciada pela AWS, o usuário não precisará de permissões explícitas para essas ações, mas o usuário não deve ter nenhuma política anexada que negue essas ações para a `aws/codecommit` chave. Se você estiver usando uma chave gerenciada pelo cliente que tenha uma Conta da AWS ID definida como principal de política para essa chave, essas permissões devem ser definidas explicitamente como `allow`. Especificamente, ao criar seu primeiro repositório e atualizar as chaves do seu repositório, você não deve ter nenhuma das seguintes permissões definidas `deny` se estiver usando uma Chave gerenciada pela AWS, e deverá ser definida como `allow` se estiver usando uma chave gerenciada pelo cliente com um diretor de política:

- `"kms:Encrypt"`
- `"kms:Decrypt"`
- `"kms:ReEncrypt"` (dependendo do contexto, pode exigir `kms:ReEncryptFrom`, `kms:ReEncryptTo` ou `kms:ReEncrypt*` não configurado para negar)
- `"kms:GenerateDataKey"`
- `"kms:GenerateDataKeyWithoutPlaintext"`
- `"kms:DescribeKey"`

Se você quiser usar sua própria chave gerenciada pelo cliente, a chave deve estar disponível no Região da AWS local onde o repositório existe. CodeCommit suporta o uso de chaves gerenciadas

pelo cliente de uma e várias regiões. Embora todos os tipos de origem de material de chaves sejam compatíveis, recomendamos usar a opção padrão KMS. Se os clientes usarem a opção Repositório de chaves externo poderá haver atrasos por parte do fornecedor da loja. Além disso, CodeCommit tem os seguintes requisitos para chaves gerenciadas pelo cliente:

- CodeCommit só suporta o uso de teclas simétricas.
- O tipo de uso da chave deve ser definido como Criptografar e descriptografar.

Consulte mais informações sobre como criar chaves gerenciadas pelo cliente em [Concepts](#) e [Creating keys](#).

Para ver as informações sobre o Chave gerenciada pela AWS gerado por CodeCommit, faça o seguinte:

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação de serviço, selecione Chaves gerenciadas pela AWS. Verifique se você está conectado ao Região da AWS local em que deseja revisar as chaves.
4. Na lista de chaves de criptografia, escolha a Chave gerenciada pela AWS com o alias aws/codecommit. As informações básicas sobre o Chave pertencente à AWS são exibidas.

Você não pode alterar ou excluir isso Chave gerenciada pela AWS.

Como os algoritmos de criptografia são usados para criptografar dados do repositório

CodeCommit usa duas abordagens diferentes para criptografar dados. Objetos Git individuais com menos de 6 MB são criptografados usando AES-GCM-256, que fornece validação de integridade de dados. Objetos entre 6 MB e o máximo de 2 GB para um único blob são criptografados usando AES-CBC-256. CodeCommit sempre valida o contexto de criptografia.

Contexto de criptografia

Cada serviço integrado ao AWS KMS especifica um contexto de criptografia para as operações de criptografia e descriptografia. O contexto de criptografia significa informações adicionais autenticadas que o AWS KMS usa para verificar a integridade dos dados. Quando especificado para a operação de criptografia, ele também deve ser especificado na operação de descriptografia. Caso contrário, a decodificação falhará. CodeCommit usa o ID do CodeCommit repositório para o contexto de

criptografia. Você pode usar o `get-repository` comando ou o CodeCommit console para encontrar o ID do repositório. Pesquise o ID do CodeCommit repositório nos AWS CloudTrail registros para entender quais operações de criptografia foram realizadas em qual chave AWS KMS para criptografar ou descriptografar dados no repositório. CodeCommit

Para obter mais informações sobre AWS KMS, consulte o [Guia do AWS Key Management Service desenvolvedor](#).

Conectando-se a AWS CodeCommit repositórios com credenciais rotativas

Você pode dar aos usuários acesso aos seus AWS CodeCommit repositórios sem configurar usuários do IAM para eles ou usar uma chave de acesso e uma chave secreta. Para atribuir permissões a identidades federadas, você cria um perfil e define permissões para a perfil. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidade de terceiros \(federação\)](#) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Guia do Usuário do Centro de Identidade do AWS IAM . Você também pode configurar o acesso baseado em funções para que os usuários do IAM acessem CodeCommit repositórios em contas separadas da Amazon Web Services (uma técnica conhecida como acesso entre contas). Para obter uma explicação detalhada sobre como configurar o acesso entre várias contas de um repositório, consulte [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#).

É possível configurar o acesso para usuários que desejam ou devem ser autenticados por meio de métodos como:

- Security Assertion Markup Language (SAML)
- Autenticação multifatorial (MFA)
- Federação
- Login da Amazon
- Amazon Cognito
- Facebook
- Google
- Provedor de identidade compatível com OpenID Connect (OIDC)

 Note

As informações a seguir se aplicam somente ao uso `git-remote-codecommit` ou ao auxiliar de AWS CLI credenciais para se conectar aos CodeCommit repositórios. Como a abordagem recomendada para acesso temporário ou federado CodeCommit é a configuração `git-remote-codecommit`, este tópico fornece exemplos de uso desse utilitário. Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Você não pode usar credenciais SSH ou Git e HTTPS para se conectar a repositórios com credenciais de CodeCommit acesso rotativo ou temporário.

Não será necessário executar estas etapas se todas as seguintes condições forem verdadeiras:

- Você está conectado a uma instância do Amazon EC2.
- Você está usando o Git e o HTTPS com o auxiliar de AWS CLI credenciais para se conectar da instância do Amazon EC2 aos repositórios. CodeCommit
- A instância do Amazon EC2 tem um perfil de instância do IAM anexado que contém as permissões de acesso descritas em [Para conexões HTTPS no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI](#) ou [Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais](#).
- Você instalou e configurou o assistente de credenciais do Git na instância do Amazon EC2, conforme descrito em [Para conexões HTTPS no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI](#) ou [Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais](#).

As instâncias do Amazon EC2 que atendem aos requisitos anteriores já estão configuradas para comunicar credenciais de acesso temporário em seu CodeCommit nome.

 Note

Você pode configurar e usar `git-remote-codecommit` em instâncias do Amazon EC2.

Para dar aos usuários acesso temporário aos seus CodeCommit repositórios, conclua as etapas a seguir.

Etapa 1: Concluir os pré-requisitos

Conclua as etapas de configuração para fornecer ao usuário acesso aos seus CodeCommit repositórios usando credenciais rotativas:

- Para acesso entre contas, consulte [Passo a passo: delegar acesso entre contas da Amazon Web Services usando perfis do IAM](#) e [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#).
- Para SAML e federação, consulte [Usando o sistema de autenticação da sua organização para conceder acesso a AWS recursos](#) e [Sobre a federação baseada em AWS STS SAML 2.0](#).
- Para saber mais sobre MFA, consulte [Usar dispositivos de autenticação multifator \(MFA\) com a AWS](#) e [Criar credenciais de segurança temporárias para habilitar o acesso de usuários do IAM](#).
- Para Login com Amazon, Amazon Cognito, Facebook, Google ou qualquer provedor de identidade compatível com OIDC, consulte [Sobre a Web Identity Federation. AWS STS](#)

Use as informações em [Autenticação e controle de acesso para AWS CodeCommit](#) para especificar as CodeCommit permissões que você deseja conceder ao usuário.

Etapa 2: obter nome do perfil ou credenciais de acesso

Se você quiser que seus usuários acessem repositórios assumindo uma função, forneça aos usuários o nome de recurso da Amazon (ARN) dessa função. Caso contrário, dependendo da forma como você configura o acesso, seu usuário pode obter credenciais rotativas de uma das seguintes maneiras:

- Para acesso entre contas, chame o comando AWS CLI [assume-role](#) ou chame a API. AWS STS [AssumeRole](#)
- Para SAML, chame o AWS CLI [assume-role-with-saml](#) comando ou a API AWS STS [AssumeRoleWithSAML](#).
- Para federação, chame os [get-federation-token](#) comandos AWS CLI [assume-role](#) ou ou. AWS STS [AssumeRoleGetFederationToken](#) APIs
- Para MFA, chame o AWS CLI [get-session-token](#) comando ou a AWS STS [GetSessionToken](#) API.
- Para Login with Amazon, Amazon Cognito, Facebook, Google ou qualquer provedor de identidade compatível com OIDC, chame o comando AWS CLI [assume-role-with-web-identity](#) ou a API. AWS STS [AssumeRoleWithWebIdentity](#)

Etapa 3: instalar git-remote-codecommit e configurar o AWS CLI

É necessário configurar o computador local para usar as credenciais de acesso instalando [git-remote-codecommit](#) e configurando um perfil na AWS CLI.

1. Siga as instruções em [Configurando](#) para configurar a AWS CLI. Use o comando `aws configure` para configurar um ou mais perfis. Considere criar um perfil nomeado para usar ao se conectar aos CodeCommit repositórios usando credenciais rotativas.
2. Você pode associar as credenciais ao perfil AWS CLI nomeado do usuário de uma das seguintes formas.
 - Se você estiver assumindo uma função para acessar CodeCommit, configure um perfil nomeado com as informações necessárias para assumir essa função. Por exemplo, se você quiser assumir uma função nomeada *CodeCommitAccess* na conta 111111111111 da Amazon Web Services, você pode configurar um perfil padrão para usar ao trabalhar com outros AWS recursos e um perfil nomeado para usar ao assumir essa função. Os comandos a seguir criam um perfil nomeado chamado *CodeAccess* que assume uma função chamada *CodeCommitAccess*. O nome do usuário *Maria_Garcia* está associado à sessão e o perfil padrão é definido como a origem de suas AWS credenciais:

```
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
```

Se você quiser verificar as alterações, visualize ou edite manualmente o arquivo `~/.aws/config` (para Linux) ou o arquivo `%UserProfile%.aws\config` (para Windows) e revise as informações sob o perfil nomeado. Por exemplo, o arquivo pode ser semelhante ao seguinte:

```
[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
```


Depois de configurar seu perfil nomeado, você poderá clonar repositórios do CodeCommit com o utilitário `git-remote-codecommit` usando o perfil nomeado. Por exemplo, para excluir um repositório denominado *MyDemoRepo*:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

- Se você estiver usando a federação de identidade da web e o OpenID Connect (OIDC), configure um perfil nomeado que faça a chamada da `AssumeRoleWithWebIdentity` API AWS Security Token Service (AWS STS) em seu nome para atualizar as credenciais temporárias. Use o `aws configure set` comando ou edite manualmente o `~/.aws/credentials` arquivo (para Linux) ou o `%UserProfile%.aws\credentials` arquivo (para Windows) para adicionar um perfil AWS CLI nomeado com os valores de configuração necessários. Por exemplo, para criar um perfil que assume a *CodeCommitAccess* função e usa um arquivo de token de identidade da web `~/my-credentials/my-token-file`

```
[CodeCommitWebIdentity]
role_arn = arn:aws:iam::111111111111:role/CodeCommitAccess
web_identity_token_file=~/my-credentials/my-token-file
role_session_name = Maria_Garcia
```

Para obter mais informações, consulte [Configurar a AWS Command Line Interface](#) e [Usar um perfil do IAM na AWS CLI](#) no Guia do usuário do AWS Command Line Interface .

Etapa 4: acessar os CodeCommit repositórios

Supondo que seu usuário tenha seguido as instruções [Conexão a um repositório](#) para se conectar aos CodeCommit repositórios, o usuário então usa a funcionalidade estendida fornecida pelo `git-remote-codecommit` Git para chamar `git clone`, `git push`, `git pull` clonar, enviar e extrair dos CodeCommit repositórios aos quais ele ou ela tem acesso. Por exemplo, para clonar um repositório:

```
git clone codecommit://CodeAccess@MyDemoRepo
```

Os comandos `commit`, `push` e `pull` do Git usam sintaxe regular do Git.

Quando o usuário usa AWS CLI e especifica o perfil AWS CLI nomeado associado às credenciais de acesso rotativas, os resultados com escopo para esse perfil são retornados.

Identity and Access Management para AWS CodeCommit

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar CodeCommit os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)
- [Autenticação e controle de acesso para AWS CodeCommit](#)
- [Como AWS CodeCommit funciona com o IAM](#)
- [CodeCommit políticas baseadas em recursos](#)
- [Autorização baseada em CodeCommit tags](#)
- [CodeCommit Funções do IAM](#)
- [AWS CodeCommit exemplos de políticas baseadas em identidade](#)
- [Solução de problemas AWS CodeCommit de identidade e acesso](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte [Solução de problemas AWS CodeCommit de identidade e acesso](#)).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte [Como AWS CodeCommit funciona com o IAM](#))
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte [AWS CodeCommit exemplos de políticas baseadas em identidade](#))

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

Autenticação e controle de acesso para AWS CodeCommit

O acesso a AWS CodeCommit requer credenciais. Essas credenciais devem ter permissões para acessar AWS recursos, como CodeCommit repositórios, e seu usuário do IAM, que você usa para gerenciar suas credenciais do Git ou a chave pública SSH que você usa para fazer conexões do Git. As seções a seguir fornecem detalhes sobre como você pode usar o [AWS Identity and Access Management \(IAM\)](#) e como ajudar CodeCommit a proteger o acesso aos seus recursos:

- [Autenticação](#)
- [Controle de acesso](#)

Autenticação

Como CodeCommit os repositórios são baseados em Git e oferecem suporte à funcionalidade básica do Git, incluindo credenciais do Git, recomendamos que você use um usuário do IAM ao trabalhar com. CodeCommit Você pode acessar CodeCommit com outros tipos de identidade, mas os outros tipos de identidade estão sujeitos a limitações, conforme descrito abaixo.

Tipos de identidade:

- Usuário do IAM: um [usuário do IAM](#) é uma identidade na sua conta da Amazon Web Services com permissões personalizadas específicas. Por exemplo, um usuário do IAM pode ter permissões para criar e gerenciar credenciais do Git para acessar repositórios. CodeCommit Esse é o tipo de usuário recomendado para trabalhar com CodeCommit. Você pode usar um nome de usuário e senha do IAM para entrar em AWS páginas da Web seguras [Console de gerenciamento da AWS](#), como os [fóruns de AWS discussão](#) ou o [AWS Support Centro](#).

É possível gerar credenciais do Git ou associar chaves públicas SSH com seu usuário do IAM, ou ainda instalar e configurar o git-remote-codecommit. Essas são as maneiras mais fáceis de configurar o Git para trabalhar com seus CodeCommit repositórios. Com as [credenciais do Git](#), você gera um nome de usuário e senha estáticos no IAM. Essas credenciais são usadas para conexões HTTP com o Git e qualquer outra ferramenta de terceiros com suporte para a autenticação usando o nome de usuário e a senha do Git. Com conexões SSH, você cria arquivos de chave pública e privada em sua máquina local que o Git CodeCommit usa para autenticação SSH. Você associa a chave pública com o usuário do IAM e armazena a chave privada no computador local. O [git-remote-codecommit](#) estende o próprio Git e não requer a configuração das credenciais do Git para o usuário.

Além disso, você pode gerar [chaves de acesso](#) para cada usuário. Use as teclas de acesso ao acessar AWS serviços de forma programática, seja por meio [de um dos AWS SDKs](#) ou usando o [AWS Command Line Interface \(AWS CLI\)](#). As ferramentas de SDK e de CLI usam as chaves de acesso para o cadastramento criptográfico das suas solicitações. Se você não usa as AWS ferramentas, você mesmo deve assinar as solicitações. CodeCommit suporta o Signature Version 4, um protocolo para autenticar solicitações de API de entrada. Para obter mais informações sobre a autenticação de solicitações, consulte [Processo de cadastramento do Signature versão 4](#) na Referência geral da AWS.

- **Usuário raiz da conta Amazon Web Services** — Ao se cadastrar AWS, você fornece um endereço de e-mail e uma senha associados à sua conta da Amazon Web Services. Estas são suas credenciais raiz e elas fornecem acesso total a todos os seus recursos da AWS . Alguns CodeCommit recursos não estão disponíveis para usuários da conta root. Além disso, a única maneira de usar o Git com sua conta root é instalar e configurar git-remote-codecommit (recomendado) ou configurar o auxiliar de AWS credenciais, que está incluído no. AWS CLI Você não pode usar as credenciais do Git ou os pares de chaves público/privada SSH com o usuário da sua conta raiz. Por esses motivos, não recomendamos usar o usuário da conta root ao interagir com CodeCommit.

 Important

Por motivos de segurança, recomendamos que você use as credenciais raiz para criar somente um usuário administrador, que é um usuário do IAM com permissões totais à sua conta da AWS . Em seguida, você pode usar esse usuário administrador para criar outros usuários e funções IAM com permissões limitadas. Para mais informações, consulte [Melhores práticas do IAM](#) e [Criar um grupo e um usuário administrador](#) no Guia do usuário do IAM.

- **IAM Identity Center e usuários no IAM Identity Center** — Centro de Identidade do AWS IAM expande os recursos de AWS Identity and Access Management fornecer um local central que reúne a administração dos usuários e seu acesso aos Contas da AWS aplicativos em nuvem. Embora seja recomendado para a maioria dos usuários que trabalham com a AWS, o IAM Identity Center não fornece mecanismos para credenciais do Git ou pares de chaves SSH. Esses usuários podem instalar e configurar CodeCommit repositórios git-remote-codecommit para clonar localmente, mas nem todos os ambientes de desenvolvimento integrados (IDEs) suportam clonagem, envio ou extração. git-remote-codecommit

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM .

- Perfil do IAM: como um usuário do IAM;, um [perfil do IAM](#) é uma identidade do IAM que você pode criar na sua conta para conceder permissões específicas.

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Note

Você não pode usar as credenciais do Git ou pares de chaves pública/privada SSH com usuários federados. Além disso, preferências do usuário não estão disponíveis para usuários federados. Para obter informações sobre como configurar conexões usando acesso federado, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas, a menos que tenha permissões, não poderá criar ou acessar CodeCommit recursos. Por exemplo, você deve ter permissões para visualizar repositórios, enviar código por push, criar e gerenciar credenciais do Git e assim por diante.

As seções a seguir descrevem como gerenciar permissões para CodeCommit. Recomendamos que você leia a visão geral primeiro.

- [Visão geral do gerenciamento de permissões de acesso aos seus CodeCommit recursos](#)
- [Usando políticas baseadas em identidade \(políticas do IAM\) para CodeCommit](#)
- [CodeCommit referência de permissões](#)

Visão geral do gerenciamento de permissões de acesso aos seus CodeCommit recursos

Cada AWS recurso é de propriedade de uma conta da Amazon Web Services. As permissões para criar ou acessar um recurso são regidas por políticas de permissões. Um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções). Alguns serviços, como AWS Lambda, também oferecem suporte à anexação de políticas de permissões aos recursos.

Note

Um administrador da conta (ou usuário administrador) é um usuário com privilégios de administrador. Para obter mais informações, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Ao conceder permissões, você decide quem recebe as permissões, os recursos relacionados às permissões concedidas e as ações específicas que deseja permitir nesses recursos.

Tópicos

- [CodeCommit recursos e operações](#)
- [Informações sobre propriedade de recursos](#)
- [Gerenciar acesso aos recursos da](#)
- [Escopo de recursos em CodeCommit](#)
- [Especificação de elementos de política: recursos, ações, efeitos e entidades de segurança](#)
- [Especificar condições em uma política](#)

CodeCommit recursos e operações

Em CodeCommit, o recurso principal é um repositório. Cada recurso possui um nome de recurso da Amazon (ARN) exclusivo associado. Em uma política, você usa um Nome de recurso da Amazon (ARN) para identificar o recurso a que a política se aplica. Para obter mais informações sobre ARNs, consulte [Amazon Resource Names \(ARN\) e AWS Service Namespaces](#) no. Referência geral da Amazon Web Services CodeCommit atualmente não oferece suporte a outros tipos de recursos, chamados de sub-recursos.

A tabela a seguir descreve como especificar CodeCommit recursos.

Tipo de recurso	Formato do ARN
Repositório	arn: aws:codecommit::: <i>region account-id repository-name</i>
Todos os CodeCommit repositórios	arn:aws:codecommit:*
Todos os CodeCommit repositórios pertencentes à conta especificada na conta especificada Região da AWS	arn: aws:codecommit::: * <i>region account-id</i>

Note

A maioria dos AWS serviços trata dois pontos (:) ou uma barra invertida (/) ARNs como o mesmo caractere. No entanto, CodeCommit requer uma correspondência exata nos padrões e regras dos recursos. Ao criar padrões de eventos, certifique-se de usar os caracteres corretos do ARN para que eles correspondam à sintaxe de ARN no recurso.

Por exemplo, é possível indicar um repositório específico (*MyDemoRepo*) em sua instrução usando o ARN dele da seguinte maneira:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo"
```

Para especificar todos os repositórios que pertencem a uma conta específica, use o caractere curinga (*) da seguinte maneira:

```
"Resource": "arn:aws:codecommit:us-west-2:111111111111:*"
```

Para especificar todos os recursos, ou se uma ação de API específica não for compatível ARNs, use o caractere curinga (*) no Resource elemento da seguinte forma:

```
"Resource": "*"
```

Você também pode usar o caractere curinga (*) para especificar todos os recursos que correspondem à parte de um nome de repositório. Por exemplo, o seguinte ARN especifica qualquer CodeCommit repositório que comece com o nome MyDemo e esteja registrado na conta da Amazon Web Services no: 111111111111 us-east-2 Região da AWS

```
arn:aws:codecommit:us-east-2:111111111111:MyDemo*
```

Para obter uma lista das operações disponíveis que funcionam com os CodeCommit recursos, consulte [CodeCommit referência de permissões](#).

Informações sobre propriedade de recursos

A conta da Amazon Web Services tem os recursos criados na conta, independentemente de quem os criou. Mais especificamente, o proprietário do recurso é a conta da Amazon Web Services da [entidade principal](#) (isto é, a conta raiz, um usuário do IAM ou um perfil do IAM) que autentica a solicitação de criação de recursos. Os seguintes exemplos mostram como isso funciona:

- Se você criar um usuário do IAM em sua conta da Amazon Web Services e conceder permissões para criar CodeCommit recursos para esse usuário, o usuário poderá criar CodeCommit recursos. No entanto, sua conta da Amazon Web Services, à qual o usuário pertence, é proprietária dos CodeCommit recursos.
- Se você usar as credenciais da conta raiz da sua conta da Amazon Web Services para criar uma regra, sua conta da Amazon Web Services é a proprietária do CodeCommit recurso.
- Se você criar uma função do IAM em sua conta da Amazon Web Services com permissões para criar CodeCommit recursos, qualquer pessoa que possa assumir a função poderá criar CodeCommit recursos. Sua conta da Amazon Web Services, à qual a função pertence, é proprietária dos CodeCommit recursos.

Gerenciar acesso aos recursos da

Para gerenciar o acesso aos AWS recursos, você usa políticas de permissões. Uma política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções para criar políticas de permissões.

Note

Esta seção discute o uso do IAM no contexto de CodeCommit. Não são fornecidas informações detalhadas sobre o serviço IAM. Para obter mais informações sobre o IAM, consulte [O que é o IAM?](#) no Guia do usuário do IAM. Para obter informações sobre a sintaxe e as descrições da política do IAM, consulte a [referência da política do IAM](#) no Manual do usuário do IAM.

Políticas de permissões que são anexadas a uma identidade do IAM são chamadas de políticas baseadas em identidade (políticas do IAM). As políticas de permissões anexadas a um recurso são chamadas de políticas baseadas em recursos. Atualmente, CodeCommit oferece suporte somente a políticas baseadas em identidade (políticas do IAM).

Tópicos

- [Políticas baseadas em identidade \(políticas do IAM\)](#)
- [Políticas baseadas em recursos](#)

Políticas baseadas em identidade (políticas do IAM)

Para gerenciar o acesso aos AWS recursos, você anexa políticas de permissões às identidades do IAM. Em CodeCommit, você usa políticas baseadas em identidade para controlar o acesso aos repositórios. Por exemplo, você pode fazer o seguinte:

- Anexe uma política de permissões a um usuário ou grupo em sua conta — Para conceder a um usuário permissões para visualizar CodeCommit recursos no CodeCommit console, anexe uma política de permissões baseada em identidade a um usuário ou grupo ao qual o usuário pertença.
- Anexe uma política de permissões a um perfil (para conceder permissões entre contas): a delegação, como quando você quer conceder acesso entre contas, envolve estabelecer uma confiança entre a conta que possui o recurso (a conta confiante) e a conta que contém os usuários que precisam acessar o recurso (a conta confiável). Uma política de permissões concede ao

usuário de um perfil as permissões necessárias para realizar as tarefas pretendidas no recurso. Uma política de confiança especifica quais contas confiáveis podem conceder a seus usuários permissões para assumir a função. Para obter mais informações, consulte [Termos e conceitos do IAM](#).

Para conceder permissões entre contas, anexe uma política de permissões baseada em identidade para um perfil do IAM. Por exemplo, o administrador na Conta A pode criar uma função para conceder permissões entre contas a outra conta da Amazon Web Services (por exemplo, Conta B) ou a um AWS serviço da seguinte forma:

1. Um administrador da Conta A cria uma função do IAM e anexa uma política de permissões à função que concede permissões em recursos da Conta A.
2. Um administrador da Conta A anexa uma política de confiança à função identificando a Conta B como a entidade principal, que pode assumir a função.
3. O administrador da Conta B pode então delegar permissões para assumir a função a qualquer usuário na Conta B. Isso permite que os usuários da Conta B criem ou acessem recursos na Conta A. Se você quiser conceder a um AWS serviço permissão para assumir a função, o principal na política de confiança também pode ser um diretor de AWS serviço. Para obter mais informações, consulte Delegação, em [Termos e conceitos do IAM](#).

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte [Gerenciamento de acesso](#) no Guia do usuário do IAM.

A política de exemplo a seguir permite que um usuário crie uma ramificação em um repositório chamado *MyDemoRepo*:

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codecommit:CreateBranch"
      ],
      "Resource" : "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
    }
  ]
}
```

```
]
}
```

Para restringir as chamadas e os recursos aos quais os usuários da sua conta têm acesso, crie políticas do IAM específicas e, em seguida, anexe essas políticas a usuários do IAM. Para obter mais informações sobre como criar funções do IAM e explorar exemplos de declarações de política do IAM para CodeCommit, consulte [Exemplos de políticas de identidade gerenciadas pelo cliente](#).

Políticas baseadas em recursos

Alguns serviços, como o Amazon S3, também suportam políticas de permissões baseadas em recursos. Por exemplo, você pode anexar uma política baseada em recursos a um bucket do S3 para gerenciar as permissões de acesso a esse bucket. CodeCommit não oferece suporte a políticas baseadas em recursos, mas você pode usar tags para identificar recursos, que podem ser usados nas políticas do IAM. Para obter um exemplo de uma política baseada em tags, consulte [Políticas baseadas em identidade \(políticas do IAM\)](#).

Escopo de recursos em CodeCommit

Em CodeCommit, você pode definir o escopo de políticas e permissões baseadas em identidade para recursos, conforme descrito em [CodeCommit recursos e operações](#). No entanto, você não pode definir o escopo da permissão `ListRepositories` para um recurso. Em vez disso, você deve definir o escopo para todos os recursos (usando o curinga `*`). Caso contrário, a ação falhará.

Todas as outras CodeCommit permissões podem ser atribuídas aos recursos.

Especificação de elementos de política: recursos, ações, efeitos e entidades de segurança

Você pode criar políticas para permitir ou negar aos usuários o acesso aos recursos, ou permitir ou negar que os usuários realizem ações específicas nesses recursos. CodeCommit define um conjunto de operações públicas de API que definem como os usuários trabalham com o serviço, seja por meio do CodeCommit console SDKs, do AWS CLI, do ou chamando-os diretamente APIs. Para conceder permissões para essas operações de API, CodeCommit defina um conjunto de ações que você pode especificar em uma política.

Algumas operações de API podem exigir permissões para mais de uma ação. Para obter mais informações sobre os recursos e operações da API, consulte [CodeCommit recursos e operações](#) e [CodeCommit referência de permissões](#).

Estes são os elementos básicos de uma política:

- **Recurso:** para identificar o recurso ao qual a política se aplica, você usa um nome do recurso da Amazon (ARN). Para obter mais informações, consulte [CodeCommit recursos e operações](#).
- **Ação:** para identificar operações de recursos que deseja permitir ou negar, use palavras-chave de ação. Por exemplo, dependendo do especificado `Effect`, a `codecommit:GetBranch` permissão permite ou nega que o usuário execute a `GetBranch` operação, que obtém detalhes sobre uma ramificação em um CodeCommit repositório.
- **Efeito:** você especifica o efeito, permitir ou negar, que ocorre quando o usuário solicita a ação específica. Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso, para ter a certeza de que um usuário não consiga acessá-lo, mesmo que uma política diferente conceda acesso.
- **Principal** — Nas políticas baseadas em identidade (políticas do IAM), o único tipo de política CodeCommit compatível, o usuário ao qual a política está vinculada é o principal implícito.

Para saber mais sobre a sintaxe da política do IAM, consulte a [Referência da política do IAM](#) no Guia do usuário do IAM.

Para ver uma tabela mostrando todas as ações da CodeCommit API e os recursos aos quais elas se aplicam, consulte [CodeCommit referência de permissões](#).

Especificar condições em uma política

Ao conceder permissões, você usa a linguagem de políticas de acesso do IAM para especificar as condições em que uma política deve entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte [Condição](#) e [Gramática de políticas](#) no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Não existem chaves de condição específicas do CodeCommit. No entanto, existem chaves AWS de condição abrangentes que você pode usar conforme apropriado. Para obter uma lista completa AWS de chaves abrangentes, consulte [Chaves disponíveis para condições](#) no Guia do usuário do IAM.

Usando políticas baseadas em identidade (políticas do IAM) para CodeCommit

Os exemplos a seguir de políticas baseadas em identidade demonstram como um administrador de conta pode anexar políticas de permissões às identidades do IAM (usuários, grupos e funções) para conceder permissões para realizar operações em recursos. CodeCommit

Important

Recomendamos que você primeiro analise os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos seus CodeCommit recursos. Para obter mais informações, consulte [Visão geral do gerenciamento de permissões de acesso aos seus CodeCommit recursos](#).

Tópicos

- [Permissões necessárias para usar o CodeCommit console](#)
- [Visualizar recursos no console](#)
- [AWS políticas gerenciadas para CodeCommit](#)
- [Exemplos de política gerenciada pelo cliente](#)

O seguinte é um exemplo de uma política de permissões com base em identidade:

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*"
      ]
    }
  ]
}
```

Essa política tem uma declaração que permite ao usuário obter informações sobre o CodeCommit repositório nomeado MyDestinationRepo e todos os CodeCommit repositórios que começam com o nome MyDemo na Região. **us-east-2**

Permissões necessárias para usar o CodeCommit console

Para ver as permissões necessárias para cada operação de CodeCommit API e para obter mais informações sobre CodeCommit operações, consulte [CodeCommit referência de permissões](#).

Para permitir que os usuários usem o CodeCommit console, o administrador deve conceder a eles permissões para CodeCommit ações. Por exemplo, você pode anexar a política [AWSCodeCommitPowerUser](#) gerenciada ou equivalente a um usuário ou grupo.

Além das permissões concedidas aos usuários por políticas baseadas em identidade, CodeCommit requer permissões para ações AWS Key Management Service (AWS KMS). Um usuário do IAM não precisa de permissões Allow explícitas para essas ações, mas não deve ter uma política anexada que defina as seguintes permissões como Deny:

```
"kms:Encrypt",  
"kms:Decrypt",  
"kms:ReEncrypt",  
"kms:GenerateDataKey",  
"kms:GenerateDataKeyWithoutPlaintext",  
"kms:DescribeKey"
```

Para obter mais informações sobre criptografia e CodeCommit, consulte [AWS KMS e criptografia](#).

Visualizar recursos no console

O CodeCommit console exige a `ListRepositories` permissão para exibir uma lista de repositórios para sua conta da Amazon Web Services no Região da AWS local em que você está conectado. O console também inclui uma função `Go to resource` (Acessar recurso) para realizar uma pesquisa por recursos que diferencia letras maiúsculas de minúsculas. Essa pesquisa é realizada na sua conta da Amazon Web Services no Região da AWS local em que você está conectado. Os seguintes recursos são exibidos nos seguintes serviços:

- AWS CodeBuild: projetos de compilação
- AWS CodeCommit: repositórios
- AWS CodeDeploy: aplicativos
- AWS CodePipeline: pipelines

Para realizar essa pesquisa nos recursos em todos os serviços, você deve ter as seguintes permissões:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Os resultados não serão retornados para os recursos de um serviço se você não tiver permissões para esse serviço. Mesmo se você tiver permissões para visualizar recursos, recursos específicos não serão retornados se houver um Deny explícito para visualizar esses recursos.

AWS políticas gerenciadas para CodeCommit

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas AWS gerenciadas do que escrever políticas você mesmo. É necessário tempo e experiência para criar [políticas gerenciadas pelo cliente do IAM](#) que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas AWS gerenciadas. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre políticas AWS gerenciadas, consulte [políticas AWS gerenciadas](#) no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política `ReadOnlyAccess` AWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de perfis de trabalho, consulte [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas AWS gerenciadas concedem as permissões necessárias para casos de uso comuns. As políticas gerenciadas CodeCommit também fornecem permissões

para realizar operações em outros serviços, como IAM, Amazon SNS e Amazon CloudWatch Events, conforme exigido pelas responsabilidades dos usuários que receberam a política em questão. Por exemplo, a `AWSCodeCommitFullAccess` política é uma política de usuário de nível administrativo que permite aos usuários com essa política criar e gerenciar regras de CloudWatch eventos para repositórios (regras cujos nomes são prefixados com) e tópicos do `codecommit` Amazon SNS para notificações sobre eventos relacionados ao repositório (tópicos cujos nomes são prefixados com), bem como administrar repositórios em `codecommit` CodeCommit.

As políticas AWS gerenciadas a seguir, que você pode anexar aos usuários em sua conta, são específicas de CodeCommit.

Tópicos

- [AWS política gerenciada: `AWSCodeCommitFullAccess`](#)
- [AWS política gerenciada: `AWSCodeCommitPowerUser`](#)
- [AWS política gerenciada: `AWSCodeCommitReadOnly`](#)
- [CodeCommit políticas e notificações gerenciadas](#)
- [AWS CodeCommit políticas gerenciadas e Amazon CodeGuru Reviewer](#)
- [CodeCommit atualizações nas políticas AWS gerenciadas](#)

AWS política gerenciada: `AWSCodeCommitFullAccess`

É possível anexar a política `AWSCodeCommitFullAccess` às suas identidades do IAM. Esta política concede acesso total CodeCommit a. Aplique essa política somente aos usuários de nível administrativo aos quais você deseja conceder controle total sobre CodeCommit repositórios e recursos relacionados em sua conta da Amazon Web Services, incluindo a capacidade de excluir repositórios.

A `AWSCodeCommitFullAccess` política contém a seguinte declaração de política:

JSON

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

```
    ],
    "Resource": "*"
  },
  {
    "Sid": "CloudWatchEventsCodeCommitRulesAccess",
    "Effect": "Allow",
    "Action": [
      "events:DeleteRule",
      "events:DescribeRule",
      "events:DisableRule",
      "events:EnableRule",
      "events:PutRule",
      "events:PutTargets",
      "events:RemoveTargets",
      "events:ListTargetsByRule"
    ],
    "Resource": "arn:aws:events:*:*:rule/codecommit*"
  },
  {
    "Sid": "SNSTopicAndSubscriptionAccess",
    "Effect": "Allow",
    "Action": [
      "sns:CreateTopic",
      "sns:DeleteTopic",
      "sns:Subscribe",
      "sns:Unsubscribe",
      "sns:SetTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:codecommit*"
  },
  {
    "Sid": "SNSTopicAndSubscriptionReadAccess",
    "Effect": "Allow",
    "Action": [
      "sns:ListTopics",
      "sns:ListSubscriptionsByTopic",
      "sns:GetTopicAttributes"
    ],
    "Resource": "*"
  },
  {
    "Sid": "LambdaReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
```

```

        "lambda:ListFunctions"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListUsers"
    ],
    "Resource": "*"
},
{
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
        "iam:ListAccessKeys",
        "iam:ListSSHPublicKeys",
        "iam:ListServiceSpecificCredentials"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "IAMUserSSHKeys",
    "Effect": "Allow",
    "Action": [
        "iam:DeleteSSHPublicKey",
        "iam:GetSSHPublicKey",
        "iam:ListSSHPublicKeys",
        "iam:UpdateSSHPublicKey",
        "iam:UploadSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "IAMSelfManageServiceSpecificCredentials",
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceSpecificCredential",
        "iam:UpdateServiceSpecificCredential",
        "iam:DeleteServiceSpecificCredential",
        "iam:ResetServiceSpecificCredential"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
}

```

```
    },
    {
      "Sid": "CodeStarNotificationsReadWriteAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "codestar-notifications:NotificationsForResource":
            "arn:aws:iam::*:role/Service*"
        }
      }
    },
    {
      "Sid": "CodeStarNotificationsListAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:SetTopicAttributes"
      ],
      "Resource": "arn:aws:sns::*:codestar-notifications*"
    },
    {
      "Sid": "AmazonCodeGuruReviewerFullAccess",
      "Effect": "Allow",
      "Action": [
```

```

        "codeguru-reviewer:AssociateRepository",
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DisassociateRepository",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerSLRCreation",
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/codeguru-
reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],

```

```

        "Resource": "*"
    },
    {
        "Sid": "CodeStarConnectionsReadOnlyAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-connections:ListConnections",
            "codestar-connections:GetConnection"
        ],
        "Resource": "arn:aws:codestar-connections:*:*:connection/*"
    }
]
}

```

AWS política gerenciada: `AWSCodeCommitPowerUser`

É possível anexar a política `AWSCodeCommitPowerUser` às suas identidades do IAM. Essa política permite que os usuários acessem todas as funcionalidades CodeCommit e recursos relacionados ao repositório, exceto que não permite que eles excluam CodeCommit repositórios nem criem ou excluam recursos relacionados ao repositório em outros serviços AWS, como o Amazon Events. CloudWatch Recomendamos que você aplique essa política à maioria dos usuários.

A `AWSCodeCommitPowerUser` política contém a seguinte declaração de política:

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codecommit:AssociateApprovalRuleTemplateWithRepository",
                "codecommit:BatchAssociateApprovalRuleTemplateWithRepositories",
                "codecommit:BatchDisassociateApprovalRuleTemplateFromRepositories",
                "codecommit:BatchGet*",
                "codecommit:BatchDescribe*",
                "codecommit:Create*",
                "codecommit>DeleteBranch",
                "codecommit>DeleteFile",
                "codecommit:Describe*",
            ]
        }
    ]
}

```



```

        "codecommit:DisassociateApprovalRuleTemplateFromRepository",
        "codecommit:EvaluatePullRequestApprovalRules",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:Merge*",
        "codecommit:OverridePullRequestApprovalRules",
        "codecommit:Put*",
        "codecommit:Post*",
        "codecommit:TagResource",
        "codecommit:Test*",
        "codecommit:UntagResource",
        "codecommit:Update*",
        "codecommit:GitPull",
        "codecommit:GitPush"
    ],
    "Resource": "*"
},
{
    "Sid": "CloudWatchEventsCodeCommitRulesAccess",
    "Effect": "Allow",
    "Action": [
        "events:DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:ListTargetsByRule"
    ],
    "Resource": "arn:aws:events:*:*:rule/codecommit*"
},
{
    "Sid": "SNSTopicAndSubscriptionAccess",
    "Effect": "Allow",
    "Action": [
        "sns:Subscribe",
        "sns:Unsubscribe"
    ],
    "Resource": "arn:aws:sns:*:*:codecommit*"
},
{
    "Sid": "SNSTopicAndSubscriptionReadAccess",
    "Effect": "Allow",

```

```
    "Action": [
      "sns:ListTopics",
      "sns:ListSubscriptionsByTopic",
      "sns:GetTopicAttributes"
    ],
    "Resource": "*"
  },
  {
    "Sid": "LambdaReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
      "lambda:ListFunctions"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
      "iam:ListUsers"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
      "iam:ListAccessKeys",
      "iam:ListSSHPublicKeys",
      "iam:ListServiceSpecificCredentials"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
  },
  {
    "Sid": "IAMUserSSHKeys",
    "Effect": "Allow",
    "Action": [
      "iam:DeleteSSHPublicKey",
      "iam:GetSSHPublicKey",
      "iam:ListSSHPublicKeys",
      "iam:UpdateSSHPublicKey",
      "iam:UploadSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
  }
```

```

    },
    {
      "Sid": "IAMSelfManageServiceSpecificCredentials",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceSpecificCredential",
        "iam:UpdateServiceSpecificCredential",
        "iam:DeleteServiceSpecificCredential",
        "iam:ResetServiceSpecificCredential"
      ],
      "Resource": "arn:aws:iam::*:user/${aws:username}"
    },
    {
      "Sid": "CodeStarNotificationsReadWriteAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "codestar-notifications:NotificationsForResource":
"arn:aws:iam::*:role/Service*"
        }
      }
    },
    {
      "Sid": "CodeStarNotificationsListAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AmazonCodeGuruReviewerFullAccess",
      "Effect": "Allow",

```

```

    "Action": [
        "codeguru-reviewer:AssociateRepository",
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DisassociateRepository",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerSLRCreation",
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/codeguru-
reviewer.amazonaws.com/AWSServiceRoleForAmazonCodeGuruReviewer",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CloudWatchEventsManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events>DeleteRule",
        "events:RemoveTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy": "codeguru-reviewer.amazonaws.com"
        }
    }
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ]
}

```

```

    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeStarConnectionsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-connections:ListConnections",
      "codestar-connections:GetConnection"
    ],
    "Resource": "arn:aws:codestar-connections:*:*:connection/*"
  }
]
}

```

AWS política gerenciada: AWSCode CommitReadOnly

É possível anexar a política AWSCodeCommitReadOnly às suas identidades do IAM. Essa política concede acesso somente para leitura CodeCommit e recursos relacionados ao repositório em outros AWS serviços, bem como a capacidade de criar e gerenciar seus próprios recursos CodeCommit relacionados (como credenciais Git e chaves SSH para o usuário do IAM usar ao acessar repositórios). Aplique essa política aos usuários a quem você deseja conceder a capacidade de ler o conteúdo de um repositório, mas não de fazer alterações a seus conteúdos.

A AWSCodeCommitReadOnly política contém a seguinte declaração de política:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGet*",
        "codecommit:BatchDescribe*",
        "codecommit:Describe*",
        "codecommit:EvaluatePullRequestApprovalRules",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:GitPull"
      ]
    }
  ]
}

```

```
    ],
    "Resource": "*"
  },
  {
    "Sid": "CloudWatchEventsCodeCommitRulesReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "events:DescribeRule",
      "events:ListTargetsByRule"
    ],
    "Resource": "arn:aws:events:*:*:rule/codecommit*"
  },
  {
    "Sid": "SNSSubscriptionAccess",
    "Effect": "Allow",
    "Action": [
      "sns:ListTopics",
      "sns:ListSubscriptionsByTopic",
      "sns:GetTopicAttributes"
    ],
    "Resource": "*"
  },
  {
    "Sid": "LambdaReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
      "lambda:ListFunctions"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMReadOnlyListAccess",
    "Effect": "Allow",
    "Action": [
      "iam:ListUsers"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMReadOnlyConsoleAccess",
    "Effect": "Allow",
    "Action": [
      "iam:ListAccessKeys",
      "iam:ListSSHPublicKeys",
```

```

        "iam:ListServiceSpecificCredentials",
        "iam:GetSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "CodeStarNotificationsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:DescribeNotificationRule"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "codestar-
notifications:NotificationsForResource": "arn:aws:codecommit:us-
east-2:111122223333:*"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonCodeGuruReviewerReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
        "codeguru-reviewer:DescribeRepositoryAssociation",
        "codeguru-reviewer:ListRepositoryAssociations",
        "codeguru-reviewer:DescribeCodeReview",
        "codeguru-reviewer:ListCodeReviews"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarConnectionsReadOnlyAccess",
    "Effect": "Allow",

```

```

        "Action": [
            "codestar-connections:ListConnections",
            "codestar-connections:GetConnection"
        ],
        "Resource": "arn:aws:codestar-connections:*:*:connection/*"
    }
}

```

CodeCommit políticas e notificações gerenciadas

AWS CodeCommit suporta notificações, que podem notificar os usuários sobre mudanças importantes nos repositórios. Políticas gerenciadas para CodeCommit incluir declarações de política para funcionalidade de notificação. Para obter mais informações, consulte [O que são notificações?](#).

Permissões relacionadas a notificações em políticas gerenciadas de acesso total

A política gerenciada `AWSCodeCommitFullAccess` inclui as declarações a seguir para permitir acesso total às notificações. Os usuários com essa política gerenciada aplicada também podem criar e gerenciar tópicos do Amazon SNS para notificações, assinar e cancelar a assinatura de usuários dos tópicos, listar tópicos a serem escolhidos como destinos para regras de notificação e listar clientes do Amazon Q Developer configurados no 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": "*"
}
}

```

Permissões relacionadas a notificações em políticas gerenciadas somente leitura

A política gerenciada `AWSCodeCommitReadOnlyAccess` inclui as instruções a seguir para permitir acesso somente leitura às notificações. Os usuários com essa política gerenciada aplicada podem visualizar notificações de recursos, mas não podem criá-los, gerenciá-los ou assiná-los.

```

{
    "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": "*"
    }
}

```

Permissões relacionadas a notificações em outras políticas gerenciadas

A política gerenciada `AWSCodeCommitPowerUser` inclui as instruções a seguir para permitir que os usuários criem, editem e assinem notificações. Os usuários não podem excluir regras de notificação nem gerenciar tags de recursos.

```

{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codecommit*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",

```

```

        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}

```

Para obter mais informações sobre o IAM e as notificações, consulte [Identity and Access Management para Notificações do AWS CodeStar](#).

AWS CodeCommit políticas gerenciadas e Amazon CodeGuru Reviewer

CodeCommit oferece suporte ao Amazon CodeGuru Reviewer, um serviço automatizado de revisão de código que usa análise de programas e aprendizado de máquina para detectar problemas comuns e recomendar correções em seu código Java ou Python. Políticas gerenciadas para CodeCommit incluir declarações de política para a funcionalidade CodeGuru Reviewer. Para obter mais informações, consulte [O que é o Amazon CodeGuru Reviewer](#).

Permissões relacionadas ao CodeGuru revisor em AWSCode CommitFullAccess

A política `AWSCodeCommitFullAccess` gerenciada inclui as seguintes declarações para permitir que o CodeGuru revisor seja associado e desassociado CodeCommit dos repositórios. Os usuários com essa política gerenciada aplicada também podem visualizar o status da associação entre os CodeCommit repositórios e o CodeGuru Reviewer e visualizar o status dos trabalhos de revisão para pull requests.

```
{
  "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"
    }
  }
}
```

Permissões relacionadas ao CodeGuru revisor em AWSCode CommitPowerUser

A política AWSCodeCommitPowerUser gerenciada inclui as seguintes declarações para permitir que os usuários associem e desassociem repositórios com o CodeGuru Reviewer, visualizem o status da associação e visualizem o status dos trabalhos de revisão para pull requests.

```
{
  "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"
  }
}
```

Permissões relacionadas ao CodeGuru revisor em AWSCode CommitReadOnly

A política `AWSCodeCommitReadOnlyAccess` gerenciada inclui as seguintes declarações para permitir acesso somente para leitura ao status da associação de CodeGuru revisores e visualizar o status dos trabalhos de revisão para pull requests. Os usuários com essa política gerenciada aplicada não podem associar ou desassociar repositórios.

```
{
  "Sid": "AmazonCodeGuruReviewerReadOnlyAccess",
  "Effect": "Allow",
  "Action": [
    "codeguru-reviewer:DescribeRepositoryAssociation",
    "codeguru-reviewer:ListRepositoryAssociations",
    "codeguru-reviewer:DescribeCodeReview",
    "codeguru-reviewer:ListCodeReviews"
  ],
  "Resource": "*"
}
```

Função vinculada ao serviço Amazon CodeGuru Reviewer

Quando você associa um repositório ao CodeGuru Reviewer, uma função vinculada ao serviço é criada para que o CodeGuru Reviewer possa detectar problemas e recomendar correções para o código Java ou Python em pull requests. A função vinculada ao serviço é chamada de `AWSServiceRoleForAmazonCodeGuruReviewer`. Para obter mais informações, consulte [Usando funções vinculadas ao serviço para o Amazon CodeGuru Reviewer](#).

Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Guia do usuário do IAM.

CodeCommit atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas CodeCommit desde que esse serviço começou a rastrear essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, assine o feed RSS em [AWS CodeCommit Histórico do documento do Guia do Usuário](#).

Alteração	Descrição	Data
AWS política gerenciada: AWSCodeCommitFullAccess e AWS política gerenciada: AWSCodeCommitPowerUser : atualização em políticas existentes	CodeCommit adicionou uma permissão a essas políticas para oferecer suporte a um tipo adicional de notificação usando o Amazon Q Developer em aplicativos de bate-papo. As AWSCodeCommitFullAccess políticas AWSCodeCommitPowerUser e foram alteradas para adicionar uma permissão, chatbot:ListMicrosoftTeamsChannelConfigurations .	16 de maio de 2023
AWS política gerenciada: AWSCodeCommitReadOnly – atualização para uma política existente	CodeCommit removeu uma permissão duplicada da política. AWSCodeCommitReadOnly Foi alterado para remover uma permissão duplicada, "iam:ListAccessKeys" .	18 de agosto de 2021
CodeCommit começou a rastrear alterações	CodeCommit começou a rastrear as mudanças em suas políticas AWS gerenciadas.	18 de agosto de 2021

Exemplos de política gerenciada pelo cliente

Você pode criar suas próprias políticas personalizadas do IAM para permitir permissões para CodeCommit ações e recursos. Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões. Você também pode criar suas próprias políticas personalizadas de IAM para integração CodeCommit entre outros AWS serviços.

Tópicos

- [Exemplos de políticas de identidade gerenciadas pelo cliente](#)

Exemplos de políticas de identidade gerenciadas pelo cliente

O exemplo a seguir de políticas do IAM concede permissões para várias CodeCommit ações. Use-os para limitar o CodeCommit acesso de seus usuários e funções do IAM. Essas políticas controlam a capacidade de realizar ações com o CodeCommit console AWS SDKs, a API ou AWS CLI o.

Note

Todos os exemplos usam a região Oeste dos EUA (Oregon) (us-west-2) e contêm uma conta fictícia. IDs

Exemplos

- [Exemplo 1: Permitir que um usuário realize CodeCommit operações em uma única Região da AWS](#)
- [Exemplo 2: permitir que um usuário use o Git para um único repositório](#)
- [Exemplo 3: permitir que um usuário conectado a partir de um intervalo de endereços IP especificado acesse um repositório](#)
- [Exemplo 4: negar ou permitir ações em ramificações](#)
- [Exemplo 5: negar ou permitir ações em repositórios com tags](#)

Exemplo 1: Permitir que um usuário realize CodeCommit operações em uma única Região da AWS

A política de permissões a seguir usa um caractere curinga ("codecommit : *") para permitir que os usuários realizem todas as CodeCommit ações na região us-east-2 e não em outra. Regiões da 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"
        }
      }
    }
  ]
}
```

Exemplo 2: permitir que um usuário use o Git para um único repositório

Em CodeCommit, as permissões da política `GitPull` do IAM se aplicam a qualquer comando do cliente Git do qual os dados são recuperados CodeCommit, incluindo `git fetch` `git clone`, e assim por diante. Da mesma forma, as permissões da política `GitPush` do IAM se aplicam a qualquer comando do cliente Git para o qual os dados são enviados. CodeCommit Por exemplo, se a permissão da política `GitPush` do IAM estiver definida como `Allow`, um usuário poderá enviar via push a exclusão de uma ramificação usando o protocolo Git. Esse envio via push não é afetado por qualquer permissão aplicada à operação `DeleteBranch` para esse usuário do IAM. A `DeleteBranch` permissão se aplica às ações realizadas com o console, o AWS CLI SDKs, o e a API, mas não com o protocolo Git.

O exemplo a seguir permite que o usuário especificado extraia e envie para o CodeCommit repositório chamado `MyDemoRepo`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codecommit:GitPull",
        "codecommit:GitPush"
      ],
      "Resource" : "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo"
    }
  ]
}
```

Exemplo 3: permitir que um usuário conectado a partir de um intervalo de endereços IP especificado acesse um repositório

Você pode criar uma política que só permite que os usuários se conectem a um repositório do CodeCommit se o endereço IP deles estiver dentro de determinado intervalo de endereços IP. Existem duas abordagens igualmente válidas para isso. Você pode criar uma Deny política que CodeCommit proíba operações se o endereço IP do usuário não estiver dentro de um bloco específico ou criar uma Allow política que permita CodeCommit operações se o endereço IP do usuário estiver dentro de um bloco específico.

Você pode criar uma política Deny que nega o acesso a todos os usuários que não estão dentro de um determinado intervalo de IPs. Por exemplo, você pode anexar a política gerenciada `AWSCodeCommitPowerUser` e uma política gerenciada pelo cliente a todos os usuários que necessitam de acesso ao seu repositório. O exemplo de política a seguir nega todas as CodeCommit permissões aos usuários cujos endereços IP não estejam dentro do bloco de endereços IP especificado de 203.0.113.0/16:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecommit:*"
      ],
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/16"
          ]
        }
      }
    }
  ]
}
```

O exemplo de política a seguir permite que o usuário especificado acesse um CodeCommit repositório nomeado MyDemoRepo com as permissões equivalentes da política AWSCodeCommitPowerUser gerenciada somente se o endereço IP estiver dentro do bloco de endereços especificado de 203.0.113.0/16:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGetRepositories",
        "codecommit:CreateBranch",
        "codecommit:CreateRepository",
        "codecommit:Get*",
        "codecommit:GitPull",

```

```

        "codecommit:GitPush",
        "codecommit:List*",
        "codecommit:Put*",
        "codecommit:Post*",
        "codecommit:Merge*",
        "codecommit:TagResource",
        "codecommit:Test*",
        "codecommit:UntagResource",
        "codecommit:Update*"
    ],
    "Resource": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "Condition": {
        "IpAddress": {
            "aws:SourceIp": [
                "203.0.113.0/16"
            ]
        }
    }
}
]
}

```

Exemplo 4: negar ou permitir ações em ramificações

Você pode criar uma política que nega aos usuários permissões para as ações que você especificar em uma ou mais ramificações. Como alternativa, você pode criar uma política que permita ações em uma ou mais ramificações que, de outra forma, elas não teriam em outras ramificações de um repositório. Você pode usar essas políticas com as políticas gerenciadas adequadas (predefinidas). Para obter mais informações, consulte [Limite os envios e fusões às ramificações em AWS CodeCommit](#).

Por exemplo, você pode criar uma Deny política que negue aos usuários a capacidade de fazer alterações em uma ramificação chamada main, incluindo a exclusão dessa ramificação, em um repositório chamado *MyDemoRepo*. Você pode usar essa política com a política gerenciada `AWSCodeCommitPowerUser`. Os usuários com essas duas políticas aplicadas poderiam criar e excluir ramificações, criar pull requests e todas as outras ações conforme permitido `AWSCodeCommitPowerUser`, mas não conseguiriam enviar alterações para a ramificação chamada main, adicionar ou editar um arquivo na ramificação principal no CodeCommit console ou mesclar ramificações ou uma pull request na ramificação principal. Como Deny é aplicado ao `GitPush`, você

deve incluir uma instrução `Null` à política para permitir que as chamadas `GitPush` iniciais sejam analisadas para validação quando os usuários fizerem envios por push de seus repositórios locais.

 Tip

Se desejar criar uma política que se aplica a todas as ramificações chamadas `master` em todos os repositórios da sua conta da Amazon Web Services, em `Resource`, especifique um asterisco (`*`) em vez de um ARN de repositório.

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"
        }
      }
    }
  ]
}
```

O exemplo de política a seguir permite ao usuário fazer alterações em uma ramificação chamada master em todos os repositórios de uma conta da Amazon Web Services. Ele não permitirá alterações em nenhuma outra ramificação. Você pode usar essa política com a política `AWSCodeCommitReadOnly` gerenciada para permitir envios automatizados para o repositório na ramificação principal. Como o efeito é `Allow`, este exemplo de política não funciona com políticas gerenciadas, como a `AWSCodeCommitPowerUser`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPush",
        "codecommit:Merge*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEqualsIfExists": {
          "codecommit:References": [
            "refs/heads/main"
          ]
        }
      }
    }
  ]
}
```

Exemplo 5: negar ou permitir ações em repositórios com tags

Você pode criar uma política que permita ou negue ações em repositórios com base nas AWS tags associadas a esses repositórios e, em seguida, aplicar essas políticas aos grupos do IAM que você configura para gerenciar usuários do IAM. Por exemplo, você pode criar uma política que negue todas as CodeCommit ações em qualquer repositório com a AWS tag Status da chave e o valor da chave Secret e, em seguida, aplicar essa política ao grupo do IAM que você criou para desenvolvedores gerais (*Developers*). Em seguida, você precisa garantir que os desenvolvedores que trabalham nesses repositórios marcados não sejam membros desse *Developers* grupo

geral, mas pertençam a um grupo diferente do IAM que não tenha a política restritiva aplicada () SecretDevelopers.

O exemplo a seguir nega todas as CodeCommit ações em repositórios marcados com o status da chave e o valor da chave 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"
    }
  }
}
]
}

```

Você pode refinar ainda mais essa estratégia, fornecendo repositórios específicos, em vez de todos os repositórios, como recursos. Você também pode criar políticas que permitam CodeCommit ações em todos os repositórios que não estejam marcados com tags específicas. Por exemplo, a política a seguir permite o equivalente a `AWSCodeCommitPowerUser` permissões para CodeCommit ações, exceto que ela só permite CodeCommit ações em repositórios não marcados com as tags especificadas:

 Note

Este exemplo de política inclui somente ações para CodeCommit. Ela não inclui ações para outros AWS serviços incluídos na política `AWSCodeCommitPowerUser` gerenciada. Para obter mais informações, consulte [AWS política gerenciada: AWSCodeCommitPowerUser](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:Associate*",
        "codecommit:Batch*",
        "codecommit:CancelUploadArchive",
        "codecommit:CreateBranch",
        "codecommit:CreateCommit",
        "codecommit:CreatePullRequest*",
        "codecommit:CreateRepository",
        "codecommit:CreateUnreferencedMergeCommit",
        "codecommit>DeleteBranch",
        "codecommit>DeleteCommentContent",
        "codecommit>DeleteFile",
        "codecommit>DeletePullRequest*",
        "codecommit:Describe*",
        "codecommit:DisassociateApprovalRuleTemplateFromRepository",
        "codecommit:EvaluatePullRequestApprovalRules",
        "codecommit:GetBlob",
        "codecommit:GetBranch",
        "codecommit:GetComment*",
        "codecommit:GetCommit",
        "codecommit:GetDifferences*",
        "codecommit:GetFile",
        "codecommit:GetFolder",
        "codecommit:GetMerge*",
        "codecommit:GetObjectIdentifier",
        "codecommit:GetPullRequest*",
        "codecommit:GetReferences",
        "codecommit:GetRepository*",

```

```

        "codecommit:GetTree",
        "codecommit:GetUploadArchiveStatus",
        "codecommit:Git*",
        "codecommit:ListAssociatedApprovalRuleTemplatesForRepository",
        "codecommit:ListBranches",
        "codecommit:ListPullRequests",
        "codecommit:ListTagsForResource",
        "codecommit:Merge*",
        "codecommit:OverridePullRequestApprovalRules",
        "codecommit:Post*",
        "codecommit:Put*",
        "codecommit:TagResource",
        "codecommit:TestRepositoryTriggers",
        "codecommit:UntagResource",
        "codecommit:UpdateComment",
        "codecommit:UpdateDefaultBranch",
        "codecommit:UpdatePullRequest*",
        "codecommit:UpdateRepository*",
        "codecommit:UploadArchive"
    ],
    "Resource": "*",
    "Condition": {
        "StringNotEquals": {
            "aws:ResourceTag/Status": "Secret",
            "aws:ResourceTag/Team": "Saanvi"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "codecommit:CreateApprovalRuleTemplate",
        "codecommit:GetApprovalRuleTemplate",
        "codecommit:ListApprovalRuleTemplates",
        "codecommit:ListRepositories",
        "codecommit:ListRepositoriesForApprovalRuleTemplate",
        "codecommit:UpdateApprovalRuleTemplateContent",
        "codecommit:UpdateApprovalRuleTemplateDescription",
        "codecommit:UpdateApprovalRuleTemplateName"
    ],
    "Resource": "*"
}
]

```

```
}
```

CodeCommit referência de permissões

As tabelas a seguir listam cada operação de CodeCommit API, as ações correspondentes para as quais você pode conceder permissões e o formato do ARN do recurso a ser usado para conceder permissões. Eles CodeCommit APIs são agrupados em tabelas com base no escopo das ações permitidas por essa API. Consulte-as ao configurar o [Controle de acesso](#) e gravar políticas de permissões que você pode anexar a uma identidade do IAM (políticas baseadas em identidade).

Quando você cria uma política de permissões, você especifica as ações no campo `Action` da política. Você especifica o valor do recurso no campo `Resource` da política como um ARN, com ou sem um caractere curinga (*).

Para expressar condições em suas CodeCommit políticas, use chaves AWS de condição abrangentes. Para obter uma lista completa AWS de chaves gerais, consulte [Chaves disponíveis](#) no Guia do usuário do IAM. Para obter informações completas sobre ações, recursos e chaves de condição CodeCommit nas políticas do IAM, consulte [Ações, recursos e chaves de condição para AWS CodeCommit](#).

Note

Para especificar uma ação, use o `codecommit:` prefixo seguido do nome da operação da API (por exemplo, `codecommit:GetRepository` ou `codecommit>CreateRepository`).

Usar curingas

Para especificar várias ações ou recursos, use um caractere curinga (*) no seu ARN. Por exemplo, `codecommit:*` especifica todas as CodeCommit ações e `codecommit:Get*` especifica todas as CodeCommit ações que começam com a palavra. `Get` O exemplo a seguir concede acesso a todos os repositórios com nomes que começam com `MyDemo`.

```
arn:aws:codecommit:us-west-2:111111111111:MyDemo*
```

Você pode usar curingas somente com os *repository-name* recursos listados na tabela a seguir. Você não pode usar curingas com *region* ou *account-id* recursos. Para obter mais informações sobre curingas, consulte [Identificadores do IAM](#), no Guia do usuário do IAM.

Tópicos

- [Permissões necessárias para comandos do cliente Git](#)
- [Permissões para ações em ramificações](#)
- [Permissões para ações em mesclagens](#)
- [Permissões para ações em solicitações pull](#)
- [Permissões para ações em modelos de regras de aprovação](#)
- [Permissões para ações em arquivos individuais](#)
- [Permissões para ações em comentários](#)
- [Permissões para ações no código confirmado](#)
- [Permissões para ações em repositórios](#)
- [Permissões para ações em tags](#)
- [Permissões para ações em acionadores](#)
- [Permissões para ações na CodePipeline integração](#)

Permissões necessárias para comandos do cliente Git

Em CodeCommit, as permissões da política `GitPull` do IAM se aplicam a qualquer comando do cliente Git do qual os dados são recuperados CodeCommit, incluindo `git fetchgit clone`, e assim por diante. Da mesma forma, as permissões da política `GitPush` do IAM se aplicam a qualquer comando do cliente Git para o qual os dados são enviados. CodeCommit Por exemplo, se a permissão da política `GitPush` do IAM estiver definida como `Allow`, um usuário poderá enviar via push a exclusão de uma ramificação usando o protocolo Git. Esse envio via push não é afetado por qualquer permissão aplicada à operação `DeleteBranch` para esse usuário do IAM. A `DeleteBranch` permissão se aplica às ações realizadas com o console, o AWS CLI SDKs, o e a API, mas não com o protocolo Git.

`GitPull` e `GitPush` são permissões de política do IAM. Elas não são ações de API.

CodeCommit Permissões necessárias para ações para comandos do cliente Git

GitPull

Ação/Ações: `codecommit:GitPull`

Necessário para extrair informações de um CodeCommit repositório para um repositório local. Esta é apenas uma permissão de política do IAM, e não uma ação de API.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GitPush

Ação/Ações: `codecommit:Git Push`

Necessário para enviar informações de um repositório local para um CodeCommit repositório. Esta é apenas uma permissão de política do IAM, e não uma ação de API.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em ramificações

As permissões a seguir permitem ou negam ações em ramificações em CodeCommit repositórios. Essas permissões se referem somente às ações realizadas no CodeCommit console e com a CodeCommit API e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Por exemplo, o comando `git show-branch -r` exibe uma lista de ramificações remotas para um repositório e suas confirmações usando o protocolo Git. Ele não é afetado por nenhuma permissão para a CodeCommit `ListBranches` operação.

Para obter mais informações sobre como criar políticas para ramificações, consulte [Limite os envios e fusões às ramificações em AWS CodeCommit](#) e [Exemplos de política gerenciada pelo cliente](#).

CodeCommit Operações de API e permissões necessárias para ações em filiais

CreateBranch

Ação/Ações: `codecommit>CreateBranch`

Necessário para criar uma ramificação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DeleteBranch

Ação/Ações: `codecommit>DeleteBranch`

Necessário para excluir uma ramificação de um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetBranch](#)

Ação/Ações: `codecommit:GetBranch`

Necessário para obter detalhes sobre uma ramificação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[ListBranches](#)

Ação/Ações: `codecommit:ListBranches`

Necessário para obter uma lista de ramificações em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesByFastForward](#)

Ação/Ações: `codecommit:MergeBranchesByFastForward`

Necessário para mesclar duas ramificações usando a estratégia de mesclagem rápida em um repositório. CodeCommit

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesBySquash](#)

Ação/Ações: `codecommit:ListBranches`

Obrigatório para mesclar duas ramificações usando a estratégia de mesclagem de compressão em um repositório do CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[MergeBranchesByThreeWay](#)

Ação/Ações: `codecommit:ListBranches`

Obrigatório para mesclar duas ramificações usando a estratégia de mesclagem de três vias em um repositório do CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[UpdateDefaultBranch](#)

Ação/Ações: `codecommit:UpdateDefaultBranch`

Necessário para alterar a ramificação padrão em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em mesclagens

As permissões a seguir permitem ou negam ações em mesclagens em CodeCommit repositórios. Essas permissões dizem respeito às ações realizadas com o CodeCommit console e a CodeCommit API, e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Para permissões relacionadas em ramificações, consulte [Permissões para ações em ramificações](#). Para permissões relacionadas em solicitações pull, consulte [Permissões para ações em solicitações pull](#).

CodeCommit Operações de API e permissões necessárias para ações de comandos de mesclagem

[BatchDescribeMergeConflicts](#)

Ação/Ações: `codecommit:BatchDescribeMergeConflicts`

Obrigatório para retornar informações sobre conflitos em uma mesclagem entre commits em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[CreateUnreferencedMergeCommit](#)

Ação/Ações: `codecommit>CreateUnreferencedMergeCommit`

Necessário criar um commit não referenciado entre duas ramificações ou commits em um CodeCommit repositório com o objetivo de compará-los e identificar possíveis conflitos.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[DescribeMergeConflicts](#)

Ação/Ações: `codecommit:DescribeMergeConflicts`

Obrigatório para retornar informações sobre conflitos de mesclagem entre as versões de base, origem e destino de um arquivo em uma possível mesclagem em um repositório do CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeCommit

Ação/Ações: `codecommit:GetMergeCommit`

Obrigatório para retornar informações sobre a mesclagem entre um commit de origem e de destino em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetMergeOptions

Ação/Ações: `codecommit:GetMergeOptions`

Obrigatório para retornar informações sobre as opções de mesclagem disponíveis entre duas ramificações ou especificadores de confirmação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em solicitações pull

As seguintes permissões permitem ou negam ações em solicitações pull em repositórios do CodeCommit. Essas permissões dizem respeito às ações realizadas com o CodeCommit console e a CodeCommit API, e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Para permissões relacionadas em comentários, consulte [Permissões para ações em comentários](#).

CodeCommit Operações de API e permissões necessárias para ações em pull requests

BatchGetPullRequests

Ação/Ações: `codecommit:BatchGetPullRequests`

Obrigatório para retornar informações sobre uma ou mais solicitações de extração em um repositório do CodeCommit. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequest

Ação/Ações: `codecommit>CreatePullRequest`

Necessário para criar uma pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreatePullRequestApprovalRule

Ação/Ações: `codecommit:CreatePullRequestApprovalRule`

Obrigatório para criar uma regra de aprovação de uma solicitação pull em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DeletePullRequestApprovalRule

Ação/Ações: `codecommit>DeletePullRequestApprovalRule`

Obrigatório para excluir uma regra de aprovação de uma solicitação pull em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

DescribePullRequestEvents

Ação/Ações: `codecommit:DescribePullRequestEvents`

Obrigatório para retornar informações sobre um ou mais eventos de pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

EvaluatePullRequestApprovalRules

Ação/Ações: `codecommit:EvaluatePullRequestApprovalRules`

Necessário para avaliar se uma pull request atendeu a todas as condições especificadas nas regras de aprovação associadas em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentsForPullRequest

Ação/Ações: `codecommit:GetCommentsForPullRequest`

Obrigatório para retornar comentários feitos em uma solicitação pull.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitsFromMergeBase

Ação/Ações: `codecommit:GetCommitsFromMergeBase`

Obrigatório para retornar informações sobre a diferença entre confirmações no contexto de uma possível mesclagem. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetMergeConflicts](#)

Ação/Ações: `codecommit:GetMergeConflicts`

Obrigatório para retornar informações sobre conflitos de mesclagem entre as ramificações de origem e de destino em uma solicitação pull.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetPullRequest](#)

Ação/Ações: `codecommit:GetPullRequest`

Obrigatório para retornar informações sobre uma solicitação pull em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetPullRequestApprovalStates](#)

Ação/Ações: `codecommit:GetPullRequestApprovalStates`

Obrigatório para retornar informações sobre os estados de aprovação de uma solicitação pull especificada.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetPullRequestOverrideState](#)

Ação/Ações: `codecommit:GetPullRequestOverrideState`

Obrigatório para retornar informações sobre se as regras de aprovação foram reservadas (substituídas) de uma solicitação pull. Em caso afirmativo, o nome do recurso da Amazon (ARN) do usuário ou identidade que substituiu as regras e seus requisitos da solicitação pull.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

ListPullRequests

Ação/Ações: `codecommit:ListPullRequests`

Obrigatório para listar solicitações pull em um repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

MergePullRequestByFastForward

Ação/Ações: `codecommit:MergePullRequestByFastForward`

Obrigatório para fechar uma solicitação pull e tentar mesclar a ramificação de origem com a ramificação de destino de uma solicitação pull usando a opção de mesclagem de avanço rápido.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

MergePullRequestBySquash

Ação/Ações: `codecommit:MergePullRequestBySquash`

Obrigatório para fechar uma solicitação pull e tentar mesclar a ramificação de origem com a ramificação de destino de uma solicitação pull usando a estratégia de mesclagem de compressão.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

MergePullRequestByThreeWay

Ação/Ações: `codecommit:MergePullRequestByThreeWay`

Obrigatório para fechar uma solicitação pull e tentar mesclar a ramificação de origem com a ramificação de destino de uma solicitação pull usando a estratégia de mesclagem de três vias.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

OverridePullRequestApprovalRules

Ação/Ações: `codecommit:OverridePullRequestApprovalRules`

Obrigatório para deixar de lado todos os requisitos da regra de aprovação para uma pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PostCommentForPullRequest

Ação/Ações: `codecommit:PostCommentForPullRequest`

Obrigatório para publicar um comentário em uma solicitação de extração em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestApprovalRuleContent

Ação/Ações: `codecommit:UpdatePullRequestApprovalRuleContent`

Necessário para alterar a estrutura de uma regra de aprovação para uma pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestApprovalState

Ação/Ações: `codecommit:UpdatePullRequestApprovalState`

Obrigatório para atualizar o estado de uma aprovação em uma solicitação pull em um repositório do CodeCommit .

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestDescription

Ação/Ações: `codecommit:UpdatePullRequestDescription`

Obrigatório para alterar a descrição de uma solicitação de extração em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestStatus

Ação/Ações: `codecommit:UpdatePullRequestStatus`

Obrigatório para alterar o status de uma solicitação de extração em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdatePullRequestTitle

Ação/Ações: `codecommit:UpdatePullRequestTitle`

Obrigatório para alterar o título de uma solicitação de extração em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em modelos de regras de aprovação

As permissões a seguir permitem ou negam ações em modelos de regras de aprovação em repositórios do CodeCommit. Essas permissões se referem somente às ações realizadas no CodeCommit console, na CodeCommit API e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Para permissões relacionadas em solicitações pull, consulte [Permissões para ações em solicitações pull](#).

CodeCommit Operações de API e permissões necessárias para ações em modelos de regras de aprovação

AssociateApprovalRuleTemplateWithRepository

Ação/Ações: `codecommit:AssociateApprovalRuleTemplateWithRepository`

Obrigatório para associar um modelo a um repositório especificado em uma conta da Amazon Web Services. Depois da associação, isso cria automaticamente regras de aprovação que correspondem às condições do modelo em cada solicitação pull criada no repositório especificado.

Recurso: *

BatchAssociateApprovalRuleTemplateWithRepositories

Ação/Ações: `codecommit:BatchAssociateApprovalRuleTemplateWithRepositories`

Obrigatório para associar um modelo a um ou mais repositórios especificados em uma conta da Amazon Web Services.

Recurso: *

[BatchDisassociateApprovalRuleTemplateFromRepositories](#)

Ação/Ações:

`codecommit:BatchDisassociateApprovalRuleTemplateFromRepositories`

Obrigatório para desassociar um modelo de um ou mais repositórios especificados em uma conta da Amazon Web Services.

Recurso: *

[CreateApprovalRuleTemplate](#)

Ação/Ações: `codecommit>CreateApprovalRuleTemplate`

Obrigatório para criar um modelo para regras de aprovação que podem ser associadas a um ou mais repositórios em sua conta da Amazon Web Services.

Recurso: *

[DeleteApprovalRuleTemplate](#)

Ação/Ações: `codecommit>DeleteApprovalRuleTemplate`

Obrigatório para excluir um modelo de regra de aprovação de uma conta da AWS .

Recurso: *

[DisassociateApprovalRuleTemplateFromRepository](#)

Ação/Ações: `codecommit:DisassociateApprovalRuleTemplateFromRepository`

Obrigatório para desassociar o modelo especificado de um repositório em uma conta da Amazon Web Services. Ele não remove regras de aprovação em solicitações pull já criadas com o modelo.

Recurso: *

[GetApprovalRuleTemplate](#)

Ação/Ações: `codecommit:GetApprovalRuleTemplate`

Obrigatório para retornar informações sobre um modelo de regra de aprovação em uma conta da Amazon Web Services.

Recurso: *

[ListApprovalRuleTemplates](#)

Ação/Ações: `codecommit:ListApprovalRuleTemplates`

Obrigatório para listar modelos de regras de aprovação em uma conta da Amazon Web Services.

Recurso: *

[ListAssociatedApprovalRuleTemplatesForRepository](#)

Ação/Ações: `codecommit:ListAssociatedApprovalRuleTemplatesForRepository`

Obrigatório para listar todos os modelos de regras de aprovação associados a um repositório especificado em uma conta da Amazon Web Services.

Recurso: *

[ListRepositoriesForApprovalRuleTemplate](#)

Ação/Ações: `codecommit:ListRepositoriesForApprovalRuleTemplate`

Obrigatório para listar todos os repositórios associados a um modelo de regra de aprovação especificado em uma conta da Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateContent](#)

Ação/Ações: `codecommit:UpdateApprovalRuleTemplateContent`

Obrigatório para atualizar o conteúdo de um modelo de regra de aprovação em uma conta da Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateDescription](#)

Ação/Ações: `codecommit:UpdateApprovalRuleTemplateDescription`

Obrigatório para atualizar a descrição de um modelo de regra de aprovação em uma conta da Amazon Web Services.

Recurso: *

[UpdateApprovalRuleTemplateName](#)

Ação/Ações: `codecommit:UpdateApprovalRuleTemplateName`

Obrigatório para atualizar o nome de um modelo de regra de aprovação em uma conta da AWS .

Recurso: *

Permissões para ações em arquivos individuais

As seguintes permissões permitem ou negam ações em arquivos individuais em repositórios do CodeCommit. Essas permissões se referem somente às ações realizadas no CodeCommit console, na CodeCommit API e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Por exemplo, o comando `git push` envia arquivos novos e alterados para um repositório do CodeCommit usando o protocolo Git. Ele não é afetado por nenhuma permissão para a CodeCommit `PutFile` operação.

CodeCommit Operações de API e permissões necessárias para ações em arquivos individuais

[DeleteFile](#)

Ação/Ações: `codecommit:DeleteFile`

Necessário para excluir um arquivo especificado de uma ramificação especificada em um CodeCommit repositório a partir do CodeCommit console.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetBlob](#)

Ação/Ações: `codecommit:GetBlob`

Necessário para visualizar o conteúdo codificado de um arquivo individual em um CodeCommit repositório a partir do console.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[GetFile](#)

Ação/Ações: `codecommit:GetFile`

Necessário para visualizar o conteúdo codificado de um arquivo especificado e seus metadados em um CodeCommit repositório a partir do console. CodeCommit

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetFolder

Ação/Ações: `codecommit:GetFolder`

Necessário para visualizar o conteúdo de uma pasta especificada em um CodeCommit repositório a partir do CodeCommit console.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PutFile

Ação/Ações: `codecommit:PutFile`

Necessário para adicionar um arquivo novo ou modificado a um CodeCommit repositório a partir do CodeCommit console, CodeCommit da API ou do AWS CLI.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em comentários

As permissões a seguir permitem ou negam ações em comentários em CodeCommit repositórios. Essas permissões dizem respeito às ações realizadas com o CodeCommit console e a CodeCommit API e aos comandos executados usando o AWS CLI. Para permissões relacionadas em comentários em solicitações pull, consulte [Permissões para ações em solicitações pull](#).

CodeCommit Operações de API e permissões necessárias para ações em repositórios

DeleteCommentContent

Ação/Ações: `codecommit>DeleteCommentContent`

Obrigatório para excluir o conteúdo de um comentário feito em uma alteração, um arquivo ou uma confirmação em um repositório. Comentários não podem ser excluídos, mas o conteúdo de um comentário pode ser removido se o usuário tem essa permissão.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetComment

Ação/Ações: `codecommit:GetComment`

Obrigatório para retornar informações sobre um comentário feito em uma alteração, arquivo ou confirmação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentReactions

Ação/Ações: `codecommit:GetCommentReactions`

Obrigatório para retornar informações sobre reações de emojis a um comentário feito em uma alteração, arquivo ou confirmação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommentsForComparedCommit

Ação/Ações: `codecommit:GetCommentsForComparedCommit`

Obrigatório para retornar informações sobre comentários feitos na comparação entre dois commits em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PostCommentForComparedCommit

Ação/Ações: `codecommit:PostCommentForComparedCommit`

Obrigatório para comentar sobre a comparação entre duas confirmações em um repositório do CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PostCommentReply

Ação/Ações: `codecommit:PostCommentReply`

Obrigatório para criar uma resposta a um comentário em uma comparação entre commits ou em uma pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PutCommentReaction

Ação/Ações: `codecommit:PutCommentReaction`

Obrigatório para responder a um comentário com um emoji em um commit ou em uma pull request em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdateComment

Ação/Ações: `codecommit:UpdateComment`

Obrigatório para editar um comentário em uma comparação entre confirmações ou em uma solicitação pull. Comentários só podem ser editados pelo autor do comentário.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações no código confirmado

As seguintes permissões permitem ou negam ações no código confirmado com os repositórios do CodeCommit. Essas permissões dizem respeito às ações realizadas com o CodeCommit console e a CodeCommit API, e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git. Por exemplo, o comando `git commit` cria uma confirmação para uma ramificação em um repositório usando o protocolo Git. Ele não é afetado por nenhuma permissão para a CodeCommit `CreateCommit` operação.

Negar explicitamente algumas dessas permissões pode resultar em consequências inesperadas no CodeCommit console. Por exemplo, configurar `GetTree` como `Deny` impede que os usuários naveguem pelo conteúdo de um repositório no console, mas não os impede de visualizar o conteúdo de um arquivo no repositório (se eles receberem um link para o arquivo no e-mail, por exemplo). Definir `GetBlob` como `Deny` impede os usuários de visualizar o conteúdo dos arquivos, mas não os impede de navegar pela estrutura de um repositório. Definir `GetCommit` como `Deny` impede que os usuários recuperem detalhes sobre confirmações. Definir `GetObjectIdentifier` como `Deny` bloqueia a maioria das funcionalidades da navegação de código. Se você definir todas essas três ações `Deny` em uma política, um usuário com essa política não poderá procurar código no CodeCommit console.

CodeCommit Operações de API e permissões necessárias para ações em código comprometido

BatchGetCommits

Ação/Ações: `codecommit:BatchGetCommits`

Obrigatório para retornar informações sobre uma ou mais confirmações em um repositório do CodeCommit. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CreateCommit

Ação/Ações: `codecommit:CreateCommit`

Obrigatório para criar uma confirmação.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommit

Ação/Ações: `codecommit:GetCommit`

Obrigatório para retornar informações sobre uma confirmação.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommitHistory

Ação/Ações: `codecommit:GetCommitHistory`

Obrigatório para retornar informações sobre o histórico de confirmações em um repositório. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetDifferences

Ação/Ações: `codecommit:GetDifferences`

Obrigatório para retornar informações sobre as diferenças em um especificador de confirmação (como ramificação, tag, HEAD, ID de confirmação ou outra referência totalmente qualificada).

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetObjectIdentifier

Ação/Ações: `codecommit:GetObjectIdentifier`

Obrigatório para resolver blobs, árvores e confirmações ao respectivo identificador. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetReferences

Ação/Ações: `codecommit:GetReferences`

Obrigatório para retornar todas as referências, como ramificações e tags. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetTree

Ação/Ações: `codecommit:GetTree`

Necessário para visualizar o conteúdo de uma árvore especificada em um CodeCommit repositório a partir do CodeCommit console. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em repositórios

As permissões a seguir permitem ou negam ações em CodeCommit repositórios. Essas permissões dizem respeito às ações realizadas com o CodeCommit console e a CodeCommit API e aos comandos executados usando o AWS CLI. Elas não pertencem a ações semelhantes que podem ser realizadas usando o protocolo Git.

CodeCommit Operações de API e permissões necessárias para ações em repositórios

[BatchGetRepositories](#)

Ação/Ações: `codecommit:BatchGetRepositories`

Necessário para obter informações sobre vários CodeCommit repositórios que estão em uma conta da Amazon Web Services. Em `Resource`, você deve especificar os nomes de todos os CodeCommit repositórios para os quais um usuário tem permissão (ou negação) de informações.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[CreateRepository](#)

Ação/Ações: `codecommit>CreateRepository`

Necessário para criar um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

[DeleteRepository](#)

Ação/Ações: `codecommit>DeleteRepository`

Necessário para excluir um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetRepository

Ação/Ações: `codecommit:GetRepository`

Necessário para obter informações sobre um único CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

ListRepositories

Ação/Ações: `codecommit:ListRepositories`

Necessário para obter uma lista dos nomes e IDs do sistema de vários CodeCommit repositórios para uma conta da Amazon Web Services. O único valor permitido de Resource para essa ação são todos os repositórios (*).

Recurso: *

UpdateRepositoryDescription

Ação/Ações: `codecommit:UpdateRepositoryDescription`

Necessário para alterar a descrição de um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UpdateRepositoryName

Ação/Ações: `codecommit:UpdateRepositoryName`

Necessário para alterar o nome de um CodeCommit repositório. Em Resource, você deve especificar os CodeCommit repositórios que podem ser alterados e os novos nomes dos repositórios.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em tags

As permissões a seguir permitem ou negam ações em AWS tags para CodeCommit recursos.

CodeCommit Operações de API e permissões necessárias para ações em tags

ListTagsForResource

Ação/Ações: `codecommit:ListTagsForResource`

Obrigatório para retornar informações sobre AWS tags configuradas em um recurso em CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

TagResource

Ação/Ações: `codecommit:TagResource`

Necessário para adicionar ou editar AWS tags para um repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UntagResource

Ação/Ações: `codecommit:UntagResource`

Necessário para remover AWS tags de um recurso em CodeCommit.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações em acionadores

As seguintes permissões permitem ou negam ações em triggers para repositórios do CodeCommit.

CodeCommit Operações de API e permissões necessárias para ações em gatilhos

GetRepositoryTriggers

Ação/Ações: `codecommit:GetRepositoryTriggers`

Obrigatório para retornar informações sobre acionadores configurados para um repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

PutRepositoryTriggers

Ação/Ações: `codecommit:PutRepositoryTriggers`

Obrigatório para criar, editar ou excluir triggers para um repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

TestRepositoryTriggers

Ação/Ações: `codecommit:TestRepositoryTriggers`

Obrigatório para testar a funcionalidade de um trigger de repositório enviando dados para o tópico ou a função configurada para o trigger.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Permissões para ações na CodePipeline integração

CodePipeline Para usar um CodeCommit repositório em uma ação de origem para um pipeline, você deve conceder todas as permissões listadas na tabela a seguir à função de serviço de. CodePipeline Caso essas permissões não sejam definidas na função de serviço ou estejam definidas como **Deny**, o pipeline não será executado automaticamente quando uma alteração for feita ao repositório, e as alterações não poderão ser liberadas manualmente.

CodeCommit Operações de API e permissões necessárias para ações na CodePipeline integração

GetBranch

Ação/Ações: `codecommit:GetBranch`

Necessário para obter detalhes sobre uma ramificação em um CodeCommit repositório.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetCommit

Ação/Ações: `codecommit:GetCommit`

Obrigatório para retornar informações sobre uma confirmação.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

UploadArchive

Ação/Ações: `codecommit:UploadArchive`

Necessário para permitir que a função de serviço carregue as alterações do repositório em um pipeline. CodePipeline Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

GetUploadArchiveStatus

Ação/Ações: `codecommit:GetUploadArchiveStatus`

Obrigatório para determinar o status de um upload de arquivamento: se ele está em andamento, concluído, cancelado ou se ocorreu um erro. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

CancelUploadArchive

Ação/Ações: `codecommit:CancelUploadArchive`

Obrigatório para cancelar o upload de um arquivamento para um pipeline. Esta é apenas uma permissão de política do IAM e não uma ação de API que você pode chamar.

Recurso: `arn:aws:codecommit:region:account-id:repository-name`

Como AWS CodeCommit funciona com o IAM

Antes de usar o IAM para gerenciar o acesso CodeCommit, você deve entender quais recursos do IAM estão disponíveis para uso CodeCommit. Para ter uma visão de alto nível de como CodeCommit e outros AWS serviços funcionam com o IAM, consulte [AWS Serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Tópicos

- [Chaves de condição](#)
- [Exemplos](#)

Chaves de condição

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

CodeCommit define seu próprio conjunto de chaves de condição e também oferece suporte ao uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte [Chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Algumas CodeCommit ações suportam a chave de `codecommit:References` condição. Para obter um exemplo de política que usa essa chave, consulte [Exemplo 4: negar ou permitir ações em ramificações](#).

Para ver uma lista de chaves de CodeCommit condição, consulte [Chaves de condição AWS CodeCommit](#) no Guia do usuário do IAM. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS CodeCommit](#).

Exemplos

Para ver exemplos de políticas CodeCommit baseadas em identidade, consulte. [AWS CodeCommit exemplos de políticas baseadas em identidade](#)

CodeCommit políticas baseadas em recursos

CodeCommit não oferece suporte a políticas baseadas em recursos.

Autorização baseada em CodeCommit tags

Você pode anexar tags a CodeCommit recursos ou passar tags em uma solicitação para CodeCommit. Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `codecommit:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`. Para obter mais informações sobre a marcação de CodeCommit recursos, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#). Para obter mais informações sobre estratégias de marcação, consulte Recursos de [marcação AWS](#).

CodeCommit também oferece suporte a políticas baseadas em tags de sessão. Para obter mais informações, consulte [Tags de sessão](#).

Usando tags para fornecer informações de identidade em CodeCommit

CodeCommit suporta o uso de tags de sessão, que são atributos de pares de valores-chave que você transmite quando assume uma função do IAM, usa credenciais temporárias ou federa um usuário em (). AWS Security Token Service AWS STS Você também pode associar tags a um usuário do IAM. Você pode usar as informações fornecidas nessas tags para facilitar a identificação

de quem fez uma alteração ou causou um evento. CodeCommit inclui os valores das tags com os seguintes nomes de chave em CodeCommit eventos:

Nome da chave	Valor
<code>displayName</code>	O nome legível por humanos para exibir e associar ao usuário (por exemplo, Mary Major ou Saanvi Sarkar).
<code>emailAddress</code>	O endereço de e-mail que você deseja exibir e associar ao usuário (por exemplo, mary_major@example.com ou saanvi_sarkar@example.com).

Se essas informações forem fornecidas, CodeCommit inclua-as nos eventos enviados à Amazon EventBridge e à Amazon CloudWatch Events. Para obter mais informações, consulte [Monitorar eventos do CodeCommit no Amazon EventBridge e no Amazon CloudWatch Events](#).

Para usar a marcação de sessão, as funções devem ter políticas que incluam a permissão `sts:TagSession` definida como `Allow`. Se você estiver usando o acesso federado, poderá configurar as informações de nome de exibição e tag de e-mail como parte da sua configuração. Por exemplo, se você estiver usando o Azure Active Directory, poderá fornecer as seguintes informações de reivindicação:

Nome da reivindicação	Valor
<code>https://aws.amazon.com/SAML/Attributes/PrincipalTag:displayName</code>	<code>user.displayName</code>
<code>https://aws.amazon.com/SAML/Attributes/PrincipalTag:emailAddress</code>	<code>user.mail</code>

Você pode usar o AWS CLI para passar tags de sessão para `displayName` e `emailAddress` usar `AssumeRole`. Por exemplo, um usuário que deseja assumir uma função chamada *Developer*

que deseja associar seu nome *Mary Major* pode usar o `assume-role` comando semelhante ao seguinte:

```
aws sts assume-role \  
--role-arn arn:aws:iam::123456789012:role/Developer \  
--role-session-name Mary-Major \  
--tags Key=displayName,Value="Mary Major" \  
      Key=emailAddress,Value="mary_major@example.com" \  
--external-id Example987
```

Para obter mais informações, consulte [AssumeRole](#).

Você pode usar a operação `AssumeRoleWithSAML` para retornar um conjunto de credenciais temporárias que incluem as tags `displayName` e `emailAddress`. Você pode usar essas tags ao acessar repositórios de CodeCommit. Isso exige que sua empresa ou grupo já tenha integrado sua solução SAML de terceiros com AWS. Em caso afirmativo, é possível transmitir atributos SAML como tags de sessão. Por exemplo, se você quiser passar atributos de identidade para um nome de exibição e endereço de e-mail para um usuário chamado *Saanvi Sarkar* como tags de sessão:

```
<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>
```

Para obter mais informações, consulte Como [passar tags de sessão usando AssumeRoleWithSAML](#).

Você pode usar a operação `AssumeRoleWithIdentity` para retornar um conjunto de credenciais temporárias que incluem as tags `displayName` e `emailAddress`. Você pode usar essas tags ao acessar repositórios de CodeCommit. Para passar tags de sessão do OpenID Connect (OIDC), é necessário incluir as tags de sessão no JSON Web Token (JWT). Por exemplo, o token JWP decodificado usado para chamar inclui `AssumeRoleWithWebIdentity` as tags `displayName` e de `emailAddress` sessão de um usuário chamado: *Li Juan*

```
{  
  "sub": "lijuan",  
  "aud": "ac_oic_client",  
  "jti": "ZYUCeREXAMPLE",
```

```
{
  "iss": "https://xyz.com",
  "iat": 1566583294,
  "exp": 1566583354,
  "auth_time": 1566583292,
  "https://aws.amazon.com/tags": {
    "principal_tags": {
      "displayName": ["Li Juan"],
      "emailAddress": ["li_juan@example.com"],
    },
    "transitive_tag_keys": [
      "displayName",
      "emailAddress"
    ]
  }
}
```

Para obter mais informações, consulte [Passando tags de sessão usando AssumeRoleWithWebIdentity](#).

Você pode usar a operação `GetFederationToken` para retornar um conjunto de credenciais temporárias que incluem as tags `displayName` e `emailAddress`. Você pode usar essas tags ao acessar repositórios de CodeCommit. Por exemplo, para usar o AWS CLI para obter um token de federação que inclua as `emailAddress` tags `displayName` e:

```
aws sts get-federation-token \
--name my-federated-user \
--tags key=displayName,value="Nikhil Jayashankar"
key=emailAddress,value=nikhil_jayashankar@example.com
```

Para obter mais informações, consulte [Passando tags de sessão usando GetFederationToken](#).

CodeCommit Funções do IAM

Um [perfil do IAM](#) é uma entidade dentro da sua conta da Amazon Web Services que tem permissões específicas.

Usando credenciais temporárias com CodeCommit

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como [AssumeRole](#) ou [GetFederationToken](#).

CodeCommit suporta o uso de credenciais temporárias. Para obter mais informações, consulte [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#).

Perfis vinculados ao serviço

[As funções vinculadas ao serviço](#) permitem que AWS os serviços acessem recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados a serviço.

CodeCommit não usa funções vinculadas ao serviço.

Perfis de serviço

Esse atributo permite que um serviço assuma um [perfil de serviço](#) em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para esse perfil. Porém, fazer isso pode alterar a funcionalidade do serviço.

CodeCommit não usa funções de serviço.

AWS CodeCommit exemplos de políticas baseadas em identidade

Por padrão, os usuários e os perfis do IAM não têm permissão para criar ou modificar recursos do CodeCommit. Eles também não podem realizar tarefas usando a AWS API Console de gerenciamento da AWS, a AWS CLI, ou. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para obter exemplos de políticas, consulte:

- [Exemplo 1: Permitir que um usuário realize CodeCommit operações em uma única Região da AWS](#)
- [Exemplo 2: permitir que um usuário use o Git para um único repositório](#)
- [Exemplo 3: permitir que um usuário conectado a partir de um intervalo de endereços IP especificado acesse um repositório](#)
- [Exemplo 4: negar ou permitir ações em ramificações](#)
- [Exemplo 5: negar ou permitir ações em repositórios com tags](#)

- [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#)

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte [Criar políticas na guia JSON](#) no Guia do usuário do IAM.

Tópicos

- [Práticas recomendadas de política](#)
- [Usando o CodeCommit console](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)
- [Visualização CodeCommit repositories com base em tags](#)

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir CodeCommit recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.

- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o CodeCommit console

Para acessar o AWS CodeCommit console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os CodeCommit recursos em sua conta da Amazon Web Services. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis do IAM) com essa política.

Para garantir que essas entidades ainda possam usar o CodeCommit console, anexe também a seguinte política AWS gerenciada às entidades. Para obter mais informações, consulte [Adição de permissões a um usuário](#) no Manual do usuário do IAM:

Para obter mais informações, consulte [Usando políticas baseadas em identidade \(políticas do IAM\) para CodeCommit](#).

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente às ações que correspondem à operação da API que você está tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
    }
  ]
}
```

Visualização CodeCommit *repositories* com base em tags

É possível utilizar condições na política baseada em identidade para controlar o acesso aos recursos do CodeCommit com base em tags. Para obter um exemplo de política que demonstra como fazer isso, consulte [Exemplo 5: negar ou permitir ações em repositórios com tags](#).

Para obter mais informações, consulte [IAM JSON Policy Elements: Condition](#) (Elementos da política JSON do IAM: Condição) no Guia do usuário do IAM.

Solução de problemas AWS CodeCommit de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com CodeCommit um IAM.

Tópicos

- [Não estou autorizado a realizar uma ação em CodeCommit](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero visualizar minhas chaves de acesso](#)
- [Sou administrador e quero permitir que outras pessoas acessem CodeCommit](#)
- [Quero permitir que pessoas fora da minha conta da Amazon Web Services acessem meus CodeCommit recursos](#)

Não estou autorizado a realizar uma ação em CodeCommit

Se isso Console de gerenciamento da AWS indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

Para obter mais informações, consulte [Permissões necessárias para usar o CodeCommit console](#).

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o CodeCommit.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no CodeCommit. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero visualizar minhas chaves de acesso

Depois de criar suas chaves de acesso de usuário do IAM, é possível visualizar seu ID da chave de acesso a qualquer momento. No entanto, você não pode visualizar sua chave de acesso secreta novamente. Se você perder sua chave secreta, crie um novo par de chaves de acesso.

As chaves de acesso consistem em duas partes: um ID de chave de acesso (por exemplo, AKIAIOSFODNN7EXAMPLE) e uma chave de acesso secreta (por exemplo, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). Como um nome de usuário e uma senha, você deve usar o ID da chave de acesso e a chave de acesso secreta em conjunto para autenticar suas solicitações. Gerencie suas chaves de acesso de forma tão segura quanto você gerencia seu nome de usuário e sua senha.

Important

Não forneça as chaves de acesso a terceiros, mesmo que seja para ajudar a [encontrar o ID de usuário canônico](#). Ao fazer isso, você pode dar a alguém acesso permanente ao seu Conta da AWS.

Ao criar um par de chaves de acesso, você é solicitado a guardar o ID da chave de acesso e a chave de acesso secreta em um local seguro. A chave de acesso secreta só está disponível no momento em que é criada. Se você perder sua chave de acesso secreta, será necessário adicionar novas chaves de acesso para seu usuário do IAM. Você pode ter no máximo duas chaves de acesso. Se você já tiver duas, você deverá excluir um par de chaves para poder criar um novo. Para visualizar as instruções, consulte [Gerenciar chaves de acesso](#) no Guia do usuário do IAM.

Sou administrador e quero permitir que outras pessoas acessem CodeCommit

Para permitir que outras pessoas acessem CodeCommit, você deve conceder permissão às pessoas ou aplicativos que precisam de acesso. Se você estiver usando o Centro de Identidade do AWS IAM para gerenciar pessoas e aplicações, atribua conjuntos de permissões a usuários ou grupos para

definir o nível de acesso. Os conjuntos de permissões criam e atribuem automaticamente políticas do IAM aos perfis do IAM associados à pessoa ou aplicação. Para ter mais informações, consulte [Conjuntos de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Se você não estiver usando o Centro de Identidade do IAM, deverá criar entidades do IAM (usuários ou perfis) para as pessoas ou aplicações que precisam de acesso. Você deve anexar uma política à entidade que concede a eles as permissões corretas no CodeCommit. Depois que as permissões forem concedidas, forneça as credenciais ao usuário ou desenvolvedor da aplicação. Eles usarão essas credenciais para acessar AWS. Para saber mais sobre como criar grupos, políticas, permissões e usuários do IAM, consulte [Identidades do IAM](#) e [Políticas e permissões no IAM](#) no Guia do usuário do IAM.

Quero permitir que pessoas fora da minha conta da Amazon Web Services acessem meus CodeCommit recursos

Para obter mais informações, consulte [Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções](#).

Resiliência no AWS CodeCommit

A infraestrutura global da AWS se baseia em Regiões da AWS e zonas de disponibilidade. A Regiões da AWS oferece várias zonas de disponibilidade separadas e isoladas fisicamente que são conectadas com baixa latência, throughputs elevadas e em redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Existe um repositório do CodeCommit ou modelo de regra de aprovação do CodeCommit na Região da AWS em que foi criado. Para obter mais informações, consulte [Regiões e endpoints de conexão Git para AWS CodeCommit](#). Para resiliência em repositórios, você pode configurar seu cliente Git para enviar via push para dois repositórios ao mesmo tempo. Para obter mais informações, consulte [Confirmações de push em um repositório do Git adicional](#).

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura global da AWS](#).

Segurança da infraestrutura no AWS CodeCommit

Como um serviço gerenciado, o AWS CodeCommit é protegido pelos procedimentos de segurança de rede global da AWS que estão descritos no whitepaper [Amazon Web Services: Overview of Security Processes](#).

Você usa chamadas de API publicadas pela AWS para acessar o CodeCommit por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter compatibilidade com conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece compatibilidade com esses modos.

As solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

É possível chamar essas operações de API de qualquer local de rede, mas o CodeCommit oferece suporte a restrições com base no endereço IP de origem. Também é possível usar políticas do CodeCommit para controlar o acesso de endpoints específicos do Amazon Virtual Private Cloud (Amazon VPC) ou de VPCs específicas. Efetivamente, isso isola o acesso à rede para um determinado recurso do CodeCommit somente da VPC específica na rede da AWS.

Para obter mais informações, consulte:

- [Exemplo 1: Permitir que um usuário realize CodeCommit operações em uma única Região da AWS](#)
- [Exemplo 3: permitir que um usuário conectado a partir de um intervalo de endereços IP especificado acesse um repositório](#)
- [Usando AWS CodeCommit com interface VPC endpoints](#)

Como monitorar o AWS CodeCommit

O monitoramento é uma parte importante na manutenção da confiabilidade, disponibilidade e desempenho do CodeCommit e das outras soluções da AWS. A AWS fornece as ferramentas de monitoramento a seguir para observar o CodeCommit, informar quando algo está errado e agir automaticamente quando apropriado:

- O Amazon EventBridge pode ser usado para automatizar seus serviços da AWS e responder automaticamente a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos dos serviços da AWS são entregues ao EventBridge quase em tempo real. Você pode escrever regras simples para determinar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra. Para obter mais informações, consulte o [Guia do usuário do Amazon EventBridge](#) e [Monitorar eventos do CodeCommit no Amazon EventBridge e no Amazon CloudWatch Events](#).
- O Amazon CloudWatch Events entrega uma transmissão quase em tempo real de eventos do sistema que descrevem alterações nos recursos da AWS. O CloudWatch Events habilita a computação orientada a eventos automatizada, já que é possível escrever regras que monitoram determinados eventos e acionam ações automatizadas em outros serviços da AWS quando esses eventos ocorrem. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Events](#) e [Monitorar eventos do CodeCommit no Amazon EventBridge e no Amazon CloudWatch Events](#).
- O Amazon CloudWatch Logs pode monitorar, armazenar e acessar seus arquivos de log do CloudTrail e de outras origens. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando certos limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por, ou em nome de, sua conta da Amazon Web Services e entrega os arquivos de log em um bucket do Amazon S3 que você especificar. É possível identificar quais usuários e contas chamaram a AWS, o endereço IP de origem do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#) e [Registrando chamadas de AWS CodeCommit API com AWS CloudTrail](#).

Monitorar eventos do CodeCommit no Amazon EventBridge e no Amazon CloudWatch Events

Você pode monitorar eventos do AWS CodeCommit no EventBridge, que fornece um fluxo de dados em tempo real das suas aplicações, de aplicações de software como serviço (SaaS) e de serviços da AWS. O EventBridge encaminha esses dados para destinos como o AWS Lambda e o Amazon Simple Notification Service. Esses eventos são iguais aos que aparecem no Amazon CloudWatch Events, que oferece um fluxo quase em tempo real de eventos do sistema que descrevem as alterações nos recursos da AWS.

Os exemplos a seguir mostram eventos do CodeCommit.

Note

O CodeCommit permite fornecer informações de `displayName` e `emailAddress` incluídas nas tags de sessão nos eventos, se essas informações estiverem disponíveis. Para obter informações, consulte [Tags de sessão](#) e [Usando tags para fornecer informações de identidade em CodeCommit](#).

Tópicos

- [evento referenceCreated](#)
- [evento referenceUpdated](#)
- [evento referenceDeleted](#)
- [evento unreferencedMergeCommitCreated](#)
- [evento commentOnCommitCreated](#)
- [evento commentOnCommitUpdated](#)
- [evento commentOnPullRequestCreated](#)
- [evento commentOnPullRequestUpdated](#)
- [evento pullRequestCreated](#)
- [evento pullRequestSourceBranchUpdated](#)
- [evento pullRequestStatusChanged](#)
- [evento pullRequestMergeStatusUpdated](#)
- [Evento approvalRuleTemplateCreated](#)

- [Evento approvalRuleTemplateUpdated](#)
- [Evento approvalRuleTemplateDeleted](#)
- [Evento approvalRuleTemplateAssociatedWithRepository](#)
- [Evento approvalRuleTemplateDisassociatedWithRepository](#)
- [Evento approvalRuleTemplateBatchAssociatedWithRepositories](#)
- [Evento approvalRuleTemplateBatchDisassociatedFromRepositories](#)
- [Evento pullRequestApprovalRuleCreated](#)
- [Evento pullRequestApprovalRuleDeleted](#)
- [Evento pullRequestApprovalRuleOverridden](#)
- [Evento pullRequestApprovalStateChanged](#)
- [Evento pullRequestApprovalRuleUpdated](#)
- [Evento reactionCreated](#)
- [Evento reactionUpdated](#)

evento referenceCreated

Neste exemplo de evento, uma ramificação chamada myBranch foi criada em um repositório chamada 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

Neste exemplo de evento, uma ramificação chamada myBranch foi atualizada por uma mesclagem em um repositório chamado 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

Neste exemplo de evento, uma ramificação chamada myBranch foi excluída em um repositório chamado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "referenceDeleted",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
    "referenceType": "branch",
    "referenceName": "myBranch",
    "referenceFullName": "refs/heads/myBranch",
    "oldCommitId": "26a8f2EXAMPLE"
  }
}
```

evento unreferencedMergeCommitCreated

Neste exemplo de evento, uma confirmação de mesclagem não referenciada foi criada em um repositório chamado MyDemoRepo.

```
{
  "version": "0",
  "id": "01234567-EXAMPLE",
  "detail-type": "CodeCommit Repository State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "event": "unreferencedMergeCommitCreated",
    "repositoryName": "MyDemoRepo",
    "repositoryId": "12345678-1234-5678-abcd-12345678abcd",

```

```

    "commitId": "7f0103fMERGE",
    "baseCommitId": "3e5a9bf1BASE",
    "sourceCommitId": "26a8f2SOURCE",
    "destinationCommitId": "3e5983DESTINATION",
    "mergeOption": "SQUASH_MERGE",
    "conflictDetailsLevel": "LINE_LEVEL",
    "conflictResolutionStrategy": "AUTOMERGE"
  }
}

```

evento commentOnCommitCreated

Neste exemplo de evento, um usuário federado chamado Mary_Major comentou em uma confirmação. Neste exemplo, seu provedor de identidade federado configurou tags de sessão para `displayName` e `emailAddress`. Essa informação está incluída no evento.

```

{
  "version": "0",
  "id": "e9dce2e9-EXAMPLE",
  "detail-type": "CodeCommit Comment on Commit",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-09-29T20:20:39Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "beforeCommitId": "3c5dEXAMPLE",
    "repositoryId": "7dd1EXAMPLE...",
    "inReplyTo": "695bEXAMPLE...",
    "notificationBody": "A comment event occurred in the following repository:
MyDemoRepo. The display name for the user is Mary Major. The email address for
the user is mary_major@example.com. The user arn:aws:sts::123456789012:federated-
user/Mary_Major made a comment. The comment was made on the following comment ID:
463bEXAMPLE.... For more information, go to the AWS CodeCommit console at https://us-
east-2.console.aws.amazon.com/codecommit/home?region=us-east-2#/repository/MyDemoRepo/
compare/3c5dEXAMPLE...f4d5EXAMPLE#463bEXAMPLE....",
    "commentId": "463bEXAMPLE...",
    "afterCommitId": "f4d5EXAMPLE",
    "event": "commentOnCommitCreated",
    "repositoryName": "MyDemoRepo",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Mary_Major",

```

```

    "displayName": "Mary Major",
    "emailAddress": "mary_major@example.com"
  }
}

```

evento commentOnCommitUpdated

Neste exemplo de evento, um usuário que assumiu uma função chamada Admin com um nome de sessão de Mary_Major editou um comentário em uma confirmação. Neste exemplo, a função incluiu tags de sessão configuradas para displayName e emailAddress. Essa informação está incluída no evento.

```

{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Commit",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "afterCommitId": "53812581",
    "beforeCommitId": "03314446",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "commentId": "a7e5471e-EXAMPLE",
    "event": "commentOnCommitUpdated",
    "inReplyTo": "bdb07d47-EXAMPLE",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Mary
Major. The email address for the user is mary_major@example.com. The user
arn:aws:sts::123456789012:federated-user/Mary_Major updated a comment or
replied to a comment. The comment was made on the following comment ID:
bdb07d47-6fe9-47b0-a839-b93cc743b2ac:468cd1cb-2dfb-4f68-9636-8de52431d1d6.
For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/
compare/0331444646178429589969823096709582251768/.../5381258150293783361471680277136017291382?
region\u003dus-east-2",
    "repositoryId": "12345678-1234-1234-1234-123456789012",
    "repositoryName": "MyDemoRepo",
    "displayName": "Mary Major",

```

```

    "emailAddress": "mary_major@example.com"
  }
}

```

evento commentOnPullRequestCreated

Neste exemplo de evento, um usuário federado chamado Saanvi_Sarkar comentou em uma solicitação pull. Neste exemplo, seu provedor de identidade federado configurou tags de sessão para `displayName` e `emailAddress`. Essa informação está incluída no evento.

```

{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Pull Request",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "beforeCommitId": "3c5dEXAMPLE",
    "repositoryId": "7dd1EXAMPLE...",
    "inReplyTo": "695bEXAMPLE...",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Saanvi
Sarkar. The email address for the user is saanvi_sarkar@example.com. The user
arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar made a comment. The comment
was made on the following Pull Request: 201. For more information, go to the AWS
CodeCommit console https://us-east-2.console.aws.amazon.com/codecommit/home?region=us-
east-2#/repository/MyDemoRepo/pull-request/201/activity#3276EXAMPLE...",
    "commentId": "463bEXAMPLE...",
    "afterCommitId": "f4d5EXAMPLE",
    "event": "commentOnPullRequestCreated",
    "repositoryName": "MyDemoRepo",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar",
    "pullRequestId": "201",
    "displayName": "Saanvi Sarkar",
    "emailAddress": "saanvi_sarkar@example.com"
  }
}

```

evento commentOnPullRequestUpdated

Neste exemplo de evento, um usuário federado chamado Saanvi_Sarkar editou um comentário em uma solicitação pull. Neste exemplo, seu provedor de identidade federado configurou tags de sessão para `displayName` e `emailAddress`. Essa informação está incluída no evento.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Comment on Pull Request",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "afterCommitId": "96814774EXAMPLE",
    "beforeCommitId": "6031971EXAMPLE",
    "callerUserArn": "arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar",
    "commentId": "40cb52f0-EXAMPLE",
    "event": "commentOnPullRequestUpdated",
    "inReplyTo": "1285e713-EXAMPLE",
    "notificationBody": "A comment event occurred in the following AWS
CodeCommit repository: MyDemoRepo. The display name for the user is Saanvi
Sarkar. The email address for the user is saanvi_sarkar@example.com. The user
arn:aws:sts::123456789012:federated-user/Saanvi_Sarkar updated a comment or
replied to a comment. The comment was made on the following Pull Request:
1. For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/1/activity#40cb52f0-aac7-4c43-b771-601eff02EXAMPLE",
    "pullRequestId": "1",
    "repositoryId": "12345678-1234-1234-1234-123456789012",
    "repositoryName": "MyDemoRepo"
  }
}
```

evento pullRequestCreated

Neste evento de exemplo, uma solicitação pull foi criada em um repositório chamado MyDemoRepo por um usuário que assumiu uma função chamada Admin com um nome de sessão de

Mary_Major. Nenhuma informação de tag de sessão foi fornecida, para que as informações não sejam incluídas no evento.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Feb 9 2019 10:18:42 PDT ",
    "description": "An example description.",
    "destinationCommit": "12241970EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestCreated",
    "isMerged": "False",
    "lastModifiedDate": "Tue Feb 9 2019 10:18:42 PDT",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. User: arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major. Event: Created. The pull request was created with the following information: Pull Request ID as 1 and title as My Example Pull Request. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1",
    "pullRequestId": "1",
    "pullRequestStatus": "Open",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9bEXAMPLE",
    "sourceCommit": "2774290EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

evento pullRequestSourceBranchUpdated

Neste exemplo de evento, um usuário que assumiu uma função chamada Admin com um nome de sessão de Mary_Major atualizou a ramificação de origem chamada test-branch para uma solicitação pull com o ID de 1.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Feb 9 2019 10:18:42 PDT",
    "description": "An example description.",
    "destinationCommit": "7644990EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestSourceBranchUpdated",
    "isMerged": "False",
    "lastModifiedDate": "Tue Feb 9 2019 10:18:42 PDT",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:sts::123456789012:assumed-role/
Admin/Mary_Major. Event: Updated. The user updated the following pull request:
1. The pull request was updated with one or more commits to the source branch:
test-branch. For more information, go to the AWS CodeCommit console https://us-
east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Open",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9b4EXAMPLE",
    "sourceCommit": "64875001EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```



```
}
```

evento pullRequestStatusChanged

Neste exemplo de evento, um usuário que assumiu uma função chamada Admin com um nome de sessão de Mary_Major fechou uma solicitação pull com o ID de 1. A solicitação pull não foi mesclada.

```
{
  "version": "0",
  "id": "98377d67-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-02-09T07:15:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Tue Jun 18 10:34:20 PDT 2019",
    "description": "An example description.",
    "destinationCommit": "95149731EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestStatusChanged",
    "isMerged": "False",
    "lastModifiedDate": "Tue Jun 18 10:34:20 PDT 2019",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major updated the following PullRequest 1. The pull request status has been updated. The status is closed. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Closed",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9bEXAMPLE",
    "sourceCommit": "4409936EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

```
}
```

evento pullRequestMergeStatusUpdated

Neste exemplo de evento, um usuário que assumiu uma função chamada Admin com um nome de sessão de Mary_Major mesclou uma solicitação pull com o ID de 1.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-06-12T10:23:43Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "callerUserArn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
    "creationDate": "Mon Mar 11 14:42:31 PDT 2019",
    "description": "An example description.",
    "destinationCommit": "4376719EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestMergeStatusUpdated",
    "isMerged": "True",
    "lastModifiedDate": "Mon Mar 11 14:42:31 PDT 2019",
    "mergeOption": "FAST_FORWARD_MERGE",
    "notificationBody": "A pull request event occurred in the following AWS CodeCommit repository: MyDemoRepo. arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major updated the following PullRequest 1. The pull request merge status has been updated. The status is merged. For more information, go to the AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-requests/1?region\u003dus-east-2",
    "pullRequestId": "1",
    "pullRequestStatus": "Closed",
    "repositoryNames": ["MyDemoRepo"],
    "revisionId": "bdc0cb9beEXAMPLE",
    "sourceCommit": "0701696EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My Example Pull Request"
  }
}
```

```
}
```

Evento approvalRuleTemplateCreated

Neste exemplo de evento, um usuário com um nome de usuário do IAM de Mary_Major criou um modelo de regra de aprovação chamado 2-approvers-required-for-main.

```
{
  "version": "0",
  "id": "f7702227-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateCreated",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template with the following name has been
created: 2-approvers-required-for-main. The ID of the created template is: d7385967-
EXAMPLE. For more information, go to the AWS CodeCommit console.",
    "repositories": {}
  }
}
```

Evento approvalRuleTemplateUpdated

Neste exemplo de evento, um usuário com um nome de usuário do IAM de Mary_Major editou um modelo de regra de aprovação chamado 2-approvers-required-for-main. O modelo de regra de aprovação não está associado a nenhum repositório.

```
{
  "version": "0",
  "id": "66403118-EXAMPLE",
```

```

"detail-type": "CodeCommit Approval Rule Template Change",
"source": "aws.codecommit",
"account": "123456789012",
"time": "2019-11-12T23:03:30Z",
"region": "us-east-2",
"resources": [

],
"detail": {
  "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
  "approvalRuleTemplateId": "c9d2b844-EXAMPLE",
  "approvalRuleTemplateName": "2-approvers-required-for-main",
  "callerUserArn": "arn:aws:iam::123456789012:user\Mary_Major",
  "creationDate": "Tue Nov 12 23:03:06 UTC 2019",
  "event": "approvalRuleTemplateDeleted",
  "lastModifiedDate": "Tue Nov 12 23:03:20 UTC 2019",
  "notificationBody": "A approval rule template event occurred in the following AWS
CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template with the following name has been
deleted: 2-approvers-required-for-main. The ID of the updated template is: c9d2b844-
EXAMPLE. For more information, go to the AWS CodeCommit console.",
  "repositories": {}
}
}

```

Evento approvalRuleTemplateDeleted

Neste exemplo de evento, um usuário com um nome de usuário do IAM Mary_Major excluiu um modelo de regra de aprovação chamado 2-approvers-required-for-main. O modelo de regra de aprovação não está associado a nenhum repositório.

```

{
  "version": "0",
  "id": "66403118-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:03:30Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "approvalRuleTemplateContentSha256": "4f3de6632EXAMPLE",
    "approvalRuleTemplateId": "c9d2b844-EXAMPLE",

```

```

    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user\Mary_Major",
    "creationDate": "Tue Nov 12 23:03:06 UTC 2019",
    "event": "approvalRuleTemplateUpdated",
    "lastModifiedDate": "Tue Nov 12 23:03:20 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following AWS
CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template with the following name has
been updated: 2-approvers-required-for-main. The ID of the updated template is:
c9d2b844-EXAMPLE. The after rule template content SHA256 is 4f3de663EXAMPLE. For more
information, go to the AWS CodeCommit console.",
    "repositories": {}
  }
}

```

Evento approvalRuleTemplateAssociatedWithRepository

Neste exemplo de evento, um usuário com um nome de usuário do IAM de Mary_Major associou um modelo de regra de aprovação chamado 2-approvers-required-for-main com um repositório chamado MyDemoRepo.

```

{
  "version": "0",
  "id": "ea1c6d73-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateAssociatedWithRepository",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been associated with the

```

```

following repository: [MyDemoRepo]. For more information, go to the AWS CodeCommit
console.",
  "repositories": {
    "MyDemoRepo": "92ca7bf2-d878-49ed-a994-336a6cc7c574"
  }
}
}

```

Evento approvalRuleTemplateDisassociatedWithRepository

Neste exemplo de evento, um usuário com um nome de usuário do IAM de Mary_Major desassociou um modelo de regra de aprovação chamado 2-approvers-required-for-main de um repositório chamado MyDemoRepo.

```

{
  "version": "0",
  "id": "ea1c6d73-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:02:27Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "d7385967-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:02:14 UTC 2019",
    "event": "approvalRuleTemplateDisassociatedFromRepository",
    "lastModifiedDate": "Wed Nov 06 19:02:14 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been disassociated from the
following repository: [MyDemoRepo]. For more information, go to the AWS CodeCommit
console.",
    "repositories": {
      "MyDemoRepo": "92ca7bf2-d878-49ed-a994-336a6cc7c574"
    }
  }
}

```

```
}
```

Evento approvalRuleTemplateBatchAssociatedWithRepositories

Neste evento de exemplo, um usuário com um nome de usuário do IAM de `Mary_Major` associou em lote um modelo de regra de aprovação chamado `2-approvers-required-for-main` com um repositório chamado `MyDemoRepo` e outro chamado `MyTestRepo`.

```
{
  "version": "0",
  "id": "0f861e5b-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:39:09Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "c71c1fe0-EXAMPLE",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Tue Nov 12 23:38:57 UTC 2019",
    "event": "batchAssociateApprovalRuleTemplateWithRepositories",
    "lastModifiedDate": "Tue Nov 12 23:38:57 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user\Mary_Major.
Additional information: An approval rule template has been batch associated with the
following repository names: [MyDemoRepo, MyTestRepo]. For more information, go to the
AWS CodeCommit console.",
    "repositories": {
      "MyDemoRepo": "MyTestRepo"
    }
  }
}
```

Evento approvalRuleTemplateBatchDisassociatedFromRepositories

Neste evento de exemplo, um usuário com um nome de usuário do IAM de Mary_Major desassociou em lote um modelo de regra de aprovação chamado 2-approvers-required-for-main de um repositório chamado MyDemoRepo e outro chamado MyTestRepo.

```
{
  "version": "0",
  "id": "e08fc996-EXAMPLE",
  "detail-type": "CodeCommit Approval Rule Template Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-12T23:39:09Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleTemplateContentSha256": "f742eebbEXAMPLE",
    "approvalRuleTemplateId": "c71c1fe0-ff91-4db4-9a45-a86a7b6c474f",
    "approvalRuleTemplateName": "2-approvers-required-for-main",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Tue Nov 12 23:38:57 UTC 2019",
    "event": "batchDisassociateApprovalRuleTemplateFromRepositories",
    "lastModifiedDate": "Tue Nov 12 23:38:57 UTC 2019",
    "notificationBody": "A approval rule template event occurred in the following
AWS CodeCommit account: 123456789012. User: arn:aws:iam::123456789012:user/Mary_Major.
Additional information: An approval rule template has been batch disassociated from
the following repository names: [MyDemoRepo, MyTestRepo]. For more information, go to
the AWS CodeCommit console.",
    "repositories": {
      "MyDemoRepo": "MyTestRepo"
    }
  }
}
```

Evento pullRequestApprovalRuleCreated

Neste exemplo de evento, um usuário com um nome de usuário do IAM Mary_Major criou uma regra de aprovação chamada 1-approver-needed para uma solicitação pull com ID 227.

```
{
```



```

"version": "0",
"id": "ad860f12-EXAMPLE",
"detail-type": "CodeCommit Pull Request State Change",
"source": "aws.codecommit",
"account": "123456789012",
"time": "2019-11-06T19:12:19Z",
"region": "us-east-2",
"resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
],
"detail": {
    "approvalRuleContentSha256": "f742eebbEXAMPLE",
    "approvalRuleId": "0a9b5dfc-EXAMPLE",
    "approvalRuleName": "1-approver-needed",
    "author": "arn:aws:iam::123456789012:user/Mary_Major",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleCreated",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Updated. Pull request: 227. Additional information: An approval rule has been
created with the following name: 1-approver-needed. For more information, go to the
AWS CodeCommit console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/
repositories/MyDemoRepo/pull-requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
        "MyDemoRepo"
    ],
    "revisionId": "3b8cecab3EXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
}
}

```

Evento pullRequestApprovalRuleDeleted

Neste exemplo de evento, um usuário com um nome de usuário do IAM de Mary_Major excluiu uma regra de aprovação chamada 1-approver-needed para uma solicitação pull com o ID 227. Um usuário do IAM com o nome Saanvi_Sarkar originalmente criou a regra de aprovação.

```
{
  "version": "0",
  "id": "c1c3509d-EXAMPLE",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:12:19Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "approvalRuleContentSha256": "f742eebbEXAMPLE",
    "approvalRuleId": "0a9b5dfc-EXAMPLE",
    "approvalRuleName": "1-approver-needed",
    "author": "arn:aws:iam::123456789012:user/Saanvi_Sarkar",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleDeleted",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following AWS
CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/Mary_Major.
Event: Created. Pull request: 227. Additional information: An approval rule has been
deleted: 1-approver-needed was deleted. For more information, go to the AWS CodeCommit
console https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/
MyDemoRepo/pull-requests/227?region=us-east-2",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",
    "repositoryNames": [
      "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
```

```

        "sourceReference": "refs/heads/test-branch",
        "title": "My example pull request"
    }
}

```

Evento pullRequestApprovalRuleOverridden

Neste exemplo de evento, os requisitos de regra de aprovação para uma solicitação pull foram reservados (OVERRIDE) por um usuário com um nome de usuário do IAM de Mary_Major. A solicitação pull foi criada por um usuário com um nome de usuário do IAM de 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"
    }
  }
}

```

Neste exemplo de evento, os requisitos de regra de aprovação para uma solicitação pull foram restabelecidos (REVOGAR).

```

{
  "version": "0",
  "id": "2895482d-13eb-b783-270d-76588e6029fa",
  "detail-type": "CodeCommit Pull Request State Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2019-11-06T19:12:19Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "author": "arn:aws:iam::123456789012:user/Li_Juan",
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "creationDate": "Wed Nov 06 19:10:58 UTC 2019",
    "description": "An An example description.",
    "destinationCommit": "194fdf00EXAMPLE",
    "destinationReference": "refs/heads/main",
    "event": "pullRequestApprovalRuleOverridden",
    "isMerged": "False",
    "lastModifiedDate": "Wed Nov 06 19:10:58 UTC 2019",
    "notificationBody": "A pull request event occurred in the following
AWS CodeCommit repository: MyDemoRepo. User: arn:aws:iam::123456789012:user/
Mary_Major. Event: Updated. Pull request name: 227. Additional information: An
override event has occurred for the approval rules for this pull request. Override
status: REVOKE. For more information, go to the AWS CodeCommit console https://
us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MyDemoRepo/pull-
requests/227?region=us-east-2",
    "overrideStatus": "REVOKE",
    "pullRequestId": "227",
    "pullRequestStatus": "Open",

```

```

    "repositoryNames": [
      "MyDemoRepo"
    ],
    "revisionId": "3b8cecabEXAMPLE",
    "sourceCommit": "29964a17EXAMPLE",
    "sourceReference": "refs/heads/test-branch",
    "title": "My example pull request"
  }
}

```

Evento pullRequestApprovalStateChanged

Neste exemplo de evento, uma solicitação pull foi aprovada por um usuário com um nome de usuário do IAM de 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"
}
}

```

Neste exemplo de evento, uma aprovação para uma solicitação pull foi revogada por um usuário com um nome de usuário do IAM de 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"
}
}

```

Evento pullRequestApprovalRuleUpdated

Neste exemplo de evento, uma regra de aprovação para uma solicitação pull foi editada por um usuário com um nome de usuário do IAM de Mary_Major. Ela também é o usuário que criou a solicitação 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

Neste exemplo de evento, uma reação a um comentário foi adicionada por um usuário com o nome de usuário do IAM de Mary_Major.

```

{
  "version": "0",
  "id": "59fccccd8-217a-32ce-2b05-561ed68a1c42",
  "detail-type": "CodeCommit Comment Reaction Change",
  "source": "aws.codecommit",
  "account": "123456789012",
  "time": "2020-04-14T00:49:03Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo"
  ],
  "detail": {
    "callerUserArn": "arn:aws:iam::123456789012:user/Mary_Major",
    "commentId": "28930161-EXAMPLE",
    "event": "commentReactionCreated",
    "notificationBody": "A comment reaction event occurred in the following AWS
CodeCommit Repository: MyDemoRepo. The user: arn:aws:iam::123456789012:user/Mary_Major
made a comment reaction # to the comment with comment ID: 28930161-EXAMPLE",

```



```

    "reactionEmojis":["#"],
    "reactionShortcodes":[":thumbsdown:"],
    "reactionUnicode":["U+1F44E"],
    "repositoryId":"12345678-1234-5678-abcd-12345678abcd",
    "repositoryName":"MyDemoRepo"
  }
}

```

Evento reactionUpdated

Neste exemplo de evento, uma reação a um comentário foi adicionada por um usuário com o nome de usuário do IAM de Mary_Major. Os usuários só podem atualizar suas próprias reações.

```

{
  "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"
  }
}

```

```
}
```

Registrando chamadas de AWS CodeCommit API com AWS CloudTrail

CodeCommit é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em CodeCommit. CloudTrail captura todas as chamadas de API para CodeCommit eventos As, incluindo chamadas do CodeCommit console, do seu cliente Git e de chamadas de código para o. CodeCommit APIs Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para. CodeCommit Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita CodeCommit, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

CodeCommit informações em CloudTrail

CloudTrail é ativado na sua conta da Amazon Web Services quando você cria a conta. Quando a atividade ocorre em CodeCommit, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar os eventos recentes em sua conta da Amazon Web Services. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo de eventos em sua conta da Amazon Web Services, incluindo eventos para CodeCommit, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, ao criar uma trilha no console, ela é aplicada a todas as regiões da . A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)

- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

Quando o CloudTrail registro é ativado em sua conta da Amazon Web Services, as chamadas de API feitas para CodeCommit ações são rastreadas em arquivos de CloudTrail log, onde são gravadas com outros registros AWS de serviços. CloudTrail determina quando criar e gravar em um novo arquivo com base no período e no tamanho do arquivo.

Todas as CodeCommit ações são registradas CloudTrail, incluindo algumas (como `GetObjectIdentifier`) que não estão atualmente documentadas na [Referência da AWS CodeCommit API](#), mas são referenciadas como permissões de acesso e documentadas em. [CodeCommit referência de permissões](#) Por exemplo, chamadas para as ações `ListRepositories` (nas AWS CLI, `aws codecommit list-repositories`), `CreateRepository` (`aws codecommit create-repository`) e `PutRepositoryTriggers` (`aws codecommit put-repository-triggers`) geram entradas nos arquivos de CloudTrail log, bem como chamadas do cliente Git para `e. GitPull GitPush` Além disso, se você tiver um CodeCommit repositório configurado como fonte para um pipeline CodePipeline, verá chamadas para CodeCommit acessar ações de permissão, como `UploadArchive` de CodePipeline. Como é CodeCommit usado AWS Key Management Service para criptografar e descriptografar repositórios, você também verá chamadas de CodeCommit para `Encrypt` e `Decrypt` ações de nos registros. AWS KMS CloudTrail

Cada entrada de log contém informações sobre quem gerou a solicitação. As informações de identidade do usuário na entrada de log ajudam você a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário-raiz ou de usuário do IAM
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado, ou se foi feita por uma função assumida
- Se a solicitação foi feita por outro AWS serviço

Para obter mais informações, consulte [Elemento `userIdentity` do CloudTrail](#).

É possível armazenar os arquivos de log no seu bucket do Amazon S3 pelo tempo que desejar, mas também pode definir regras do ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Por padrão, os arquivos de log são criptografados com criptografia do lado do servidor (SSE) do Amazon S3.

Entendendo as entradas do arquivo de CodeCommit log

CloudTrail os arquivos de log podem conter uma ou mais entradas de log. Cada entrada lista vários eventos com formatação JSON. Um evento de log representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação, e assim por diante. As entradas de log não são um rastreamento de pilha ordenada das chamadas de API pública. Assim, elas não são exibidas em nenhuma ordem específica.

Note

Este exemplo foi formatado para melhorar a legibilidade. Em um arquivo de CloudTrail log, todas as entradas e eventos são concatenados em uma única linha. Esse exemplo também foi limitado a uma única CodeCommit entrada. Em um arquivo de CloudTrail log real, você vê entradas e eventos de vários AWS serviços.

Sumário

- [Exemplo: uma entrada de registro para listar CodeCommit repositórios](#)
- [Exemplo: uma entrada de registro para criar um CodeCommit repositório](#)
- [Exemplos: entradas de registro para chamadas pull do Git para um repositório CodeCommit](#)
- [Exemplo: uma entrada de registro para um envio bem-sucedido para um CodeCommit repositório](#)

Exemplo: uma entrada de registro para listar CodeCommit repositórios

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListRepositories` ação.

Note

Embora `ListRepositories` retorne uma lista de repositórios, as respostas não mutáveis não são registradas nos CloudTrail registros, portanto `responseElements`, são mostradas como `null` no arquivo de 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": "2016-12-14T17:57:36Z",
  "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"
}

```

Exemplo: uma entrada de registro para criar um CodeCommit repositório

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateRepository` ação na região Leste dos EUA (Ohio).

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "CreateRepository",
  "awsRegion": "us-east-2",
  "sourceIpAddress": "203.0.113.12",

```

```

"userAgent": "aws-cli/1.10.53 Python/2.7.9 Windows/8 boto3/1.4.43",
"requestParameters": {
  "repositoryDescription": "Creating a demonstration repository.",
  "repositoryName": "MyDemoRepo"
},
"responseElements": {
  "repositoryMetadata": {
    "arn": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "creationDate": "Dec 14, 2016 6:19:14 PM",
    "repositoryId": "8afe792d-EXAMPLE",
    "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryName": "MyDemoRepo",
    "accountId": "111122223333",
    "cloneUrlHttp": "https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo",
    "repositoryDescription": "Creating a demonstration repository.",
    "lastModifiedDate": "Dec 14, 2016 6:19:14 PM"
  }
},
"requestID": "d148de46-EXAMPLE",
"eventID": "740f179d-EXAMPLE",
"readOnly": false,
"resources": [
  {
    "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
    "accountId": "111122223333",
    "type": "AWS::CodeCommit::Repository"
  }
],
"eventType": "AwsApiCall",
"apiVersion": "2015-04-13",
"recipientAccountId": "111122223333"
}

```

Exemplos: entradas de registro para chamadas pull do Git para um repositório CodeCommit

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `GitPull` ação em que o repositório local já está. 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"
}

```

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `GitPull` ação em que o repositório local não está up-to-date e, portanto, os dados são transferidos do repositório para o CodeCommit repositório local.

```

{
  "eventVersion": "1.05",
  "userIdentity": {

```

```
{
  "type": "IAMUser",
  "principalId": "AIDACKCEVSQ6C2EXAMPLE",
  "arn": "arn:aws:iam::444455556666:user/Mary_Major",
  "accountId": "444455556666",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "userName": "Mary_Major"
},
{
  "eventTime": "2016-12-14T18:19:15Z",
  "eventSource": "codecommit.amazonaws.com",
  "eventName": "GitPull",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "git/2.10.1",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "protocol": "HTTP",
    "capabilities": [
      "multi_ack_detailed",
      "side-band-64k",
      "thin-pack"
    ],
    "dataTransferred": true,
    "repositoryName": "MyDemoRepo",
    "repositoryId": "8afe792d-EXAMPLE",
    "shallow": false
  },
  "requestID": "d148de46-EXAMPLE",
  "eventID": "740f179d-EXAMPLE",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:codecommit:us-east-2:111122223333:MyDemoRepo",
      "accountId": "111122223333",
      "type": "AWS::CodeCommit::Repository"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```


Exemplo: uma entrada de registro para um envio bem-sucedido para um CodeCommit repositório

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra uma `GitPush` ação bem-sucedida. A ação `GitPush` aparece duas vezes na entrada de log de um push bem-sucedido.

```
{
  "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"  
  }  
}
```

Criação de CodeCommit recursos com AWS CloudFormation

AWS CodeCommit é integrado com AWS CloudFormation, um serviço que ajuda você a modelar e configurar seus AWS recursos para que você possa gastar menos tempo criando e gerenciando seus recursos e infraestrutura. Você cria um modelo que descreve todos os AWS recursos que você deseja (como repositórios) e CloudFormation provisiona e configura esses recursos para você.

Ao usar CloudFormation, você pode reutilizar seu modelo para configurar seus CodeCommit recursos de forma consistente e repetida. Descreva seus recursos uma vez e, em seguida, provisione os mesmos recursos repetidamente em várias Contas da AWS regiões.

CodeCommit e CloudFormation modelos

Para provisionar e configurar recursos CodeCommit e serviços relacionados, você deve entender [CloudFormation os modelos](#). Os modelos são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os recursos que você deseja provisionar em suas CloudFormation pilhas. Se você não estiver familiarizado com JSON ou YAML, você pode usar o CloudFormation Designer para ajudá-lo a começar a usar modelos. CloudFormation Para obter mais informações, consulte [O que é o Designer CloudFormation ?](#) no Manual do usuário do AWS CloudFormation .

CodeCommit suporta a criação de repositórios em CloudFormation Diferentemente da criação de repositórios a partir do console ou da linha de comando, você pode usar CloudFormation para criar repositórios e enviar automaticamente o código para o repositório recém-criado a partir de um arquivo.zip especificado em um bucket do Amazon S3. Para obter mais informações, incluindo exemplos de modelos JSON e YAML para repositórios, consulte. [AWS::CodeCommit::Repository](#)

Ao criar um CodeCommit repositório usando CloudFormation, você tem a opção de enviar código para esse repositório como parte do processo de criação, desde que o arquivo tenha menos de 20 MB, configurando propriedades no [AWS::CodeCommit::Repository](#) Code. Você pode especificar o bucket do Amazon S3 em que o código é armazenado e, opcionalmente, usar a [BranchName propriedade](#) para especificar o nome da ramificação padrão que será criada na confirmação inicial desse código. Essas propriedades são usadas somente na criação inicial do repositório e são ignoradas nas atualizações de pilhas. Você não pode usar essas propriedades para fazer confirmações adicionais em um repositório ou para alterar o nome da ramificação padrão após a confirmação inicial ser feita.

Note

Em 19 de janeiro de 2021, AWS alterou o nome da ramificação padrão CodeCommit de master para main. Essa alteração de nome afeta o comportamento padrão da CodeCommit criação da confirmação inicial para repositórios usando o CodeCommit console, o CodeCommit APIs AWS SDKs, o e o. AWS CLI Os repositórios criados com CloudFormation ou AWS CDK com uma confirmação inicial de código como parte da criação se alinham a essa alteração a partir de 4 de março de 2021. Essa alteração não afeta repositórios ou ramificações existentes. Os clientes que usam clientes Git locais para criar suas confirmações iniciais têm um nome de ramificação padrão que segue a configuração desses clientes Git. Para obter mais informações, consulte [Trabalhar com ramificações](#), [Criar uma confirmação](#) e [Alterar as configurações da ramificação](#).

Você também pode criar modelos que criam recursos relacionados, como [regras de notificação](#) para repositórios, [projetos de compilação do AWS CodeBuild](#), [aplicativos do AWS CodeDeploy](#) e [pipelines do AWS CodePipeline](#).

Exemplos de modelos

Os exemplos a seguir criam um CodeCommit repositório chamado *MyDemoRepo*. O repositório recém-criado é preenchido com código armazenado em um bucket do Amazon S3 *MySourceCodeBucket* chamado e colocado em uma ramificação *development* chamada, que é a ramificação padrão do repositório.

Note

O nome do bucket do Amazon S3 que contém o arquivo ZIP com o conteúdo que será confirmado para o novo repositório pode ser especificado usando um ARN ou o nome do bucket na conta do Amazon Web Services. Essa é uma chave de objeto do Amazon S3 conforme definido no [Guia do desenvolvedor do Amazon S3](#).

JSON:

```
{
  "MyRepo": {
    "Type": "AWS::CodeCommit::Repository",
```

```
"Properties": {
  "RepositoryName": "MyDemoRepo",
  "RepositoryDescription": "This is a repository for my project with code
from MySourceCodeBucket.",
  "Code": {
    "BranchName": "development",
    "S3": {
      "Bucket": "MySourceCodeBucket",
      "Key": "MyKey",
      "ObjectVersion": "1"
    }
  }
}
```

YAML:

```
MyRepo:
  Type: AWS::CodeCommit::Repository
  Properties:
    RepositoryName: MyDemoRepo
    RepositoryDescription: This is a repository for my project with code from
MySourceCodeBucket.
  Code:
    BranchName: development
  S3:
    Bucket: MySourceCodeBucket,
    Key: MyKey,
    ObjectVersion: 1
```

Para obter mais exemplos, consulte [AWS::CodeCommit::Repository](#).

CloudFormation, CodeCommit, e o AWS Cloud Development Kit (AWS CDK)

Repositórios criados usando a CloudFormation funcionalidade de AWS CDK uso em sua criação. Entender como CloudFormation os modelos funcionam com CodeCommit os recursos pode ajudar você a criar e gerenciar seu AWS CDK código. Para obter mais informações sobre o AWS CDK, consulte o [Guia do AWS Cloud Development Kit \(AWS CDK\) desenvolvedor](#) e a [referência AWS CDK da API](#).

O exemplo de texto AWS CDK datilografado a seguir cria um CodeCommit repositório chamado. *MyDemoRepo* O repositório recém-criado é preenchido com código armazenado em um bucket do Amazon S3 *MySourceCodeBucket* chamado e colocado em uma ramificação *development* chamada, que é a ramificação padrão do repositório.

```
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"
        }
      },
    });
  }
}
```

Saiba mais sobre CloudFormation

Para saber mais sobre isso CloudFormation, consulte os seguintes recursos:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guia do usuário](#)
- [AWS CloudFormation Guia do usuário da interface de linha de comando](#)

Solução de problemas do AWS CodeCommit

As informações a seguir podem ajudar a solucionar problemas comuns no AWS CodeCommit.

Tópicos

- [Solução de problemas de credenciais do Git e conexões HTTPS para AWS CodeCommit](#)
- [Solução de problemas git-remote-codecommit e AWS CodeCommit](#)
- [Solução de problemas de conexões SSH com AWS CodeCommit](#)
- [Solução de problemas do auxiliar de credenciais e das conexões HTTPS com AWS CodeCommit](#)
- [Solução de problemas de clientes Git e AWS CodeCommit](#)
- [Solução de problemas de erros de acesso e AWS CodeCommit](#)
- [Solução de problemas de erros de configuração e AWS CodeCommit](#)
- [Solução de problemas de erros do console e AWS CodeCommit](#)
- [Solução de problemas, gatilhos e AWS CodeCommit](#)
- [Ativação da depuração](#)

Solução de problemas de credenciais do Git e conexões HTTPS para AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas comuns ao usar credenciais do Git e HTTPS para se conectar aos repositórios. AWS CodeCommit

Tópicos

- [Credenciais do Git para AWS CodeCommit: continuo vendo uma solicitação de credenciais quando me conecto ao meu CodeCommit repositório no terminal ou na linha de comando](#)
- [Credenciais do Git para AWS CodeCommit: Eu configurei as credenciais do Git, mas meu sistema não as está usando](#)

Credenciais do Git para AWS CodeCommit: continuo vendo uma solicitação de credenciais quando me conecto ao meu CodeCommit repositório no terminal ou na linha de comando

Problema: ao tentar enviar, extrair ou interagir de outra forma com um CodeCommit repositório a partir do terminal ou da linha de comando, você é solicitado a fornecer um nome de usuário e uma senha e deve fornecer as credenciais do Git para seu usuário do IAM.

Correções possíveis: as causas mais comuns para esse erro são que seu computador local está executando um sistema operacional que não é compatível com o gerenciamento de credenciais ou não tem um utilitário de gerenciamento de credenciais instalado, ou as credenciais do Git de seu usuário do IAM não foram salvas para um desses sistemas de gerenciamento de credenciais. Dependendo de seu sistema operacional e ambiente local, talvez seja necessário instalar um gerenciador de credenciais, configurar o gerenciador de credenciais incluído no sistema operacional ou personalizar o ambiente local para usar o armazenamento de credenciais. Por exemplo, se seu computador estiver executando o macOS, você pode usar o utilitário Acesso às Chaves para armazenar suas credenciais. Se seu computador for Windows, você pode usar o Git Credential Manager que é instalado com o Git para Windows. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#) e [Credential Storage](#) na documentação do Git.

Credenciais do Git para AWS CodeCommit: Eu configurei as credenciais do Git, mas meu sistema não as está usando

Problema: quando você tenta usar CodeCommit com um cliente Git, o cliente não parece usar as credenciais do Git para seu usuário do IAM.

Possíveis correções: a causa mais comum desse erro é que você configurou anteriormente seu computador para usar o auxiliar de credenciais que está incluído na AWS CLI. Verifique o arquivo .gitconfig para seções de configuração semelhantes às seguintes e remova-as:

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

Salve o arquivo e abra uma nova linha de comando ou sessão de terminal antes de tentar se conectar novamente.

Você também pode ter vários auxiliares ou gerenciadores de credenciais configurados no seu computador, e seu sistema pode estar assumindo outra configuração como padrão. Para restaurar o assistente de credenciais usado por padrão, você pode usar a opção `--system` em vez de `--global` ou `--local` ao executar o comando `git config`.

Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git e Credential Storage](#) na documentação do Git.

Solução de problemas git-remote-codecommit e AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas com o git-remote-codecommit ao conectar-se com repositórios do AWS CodeCommit .

Tópicos

- [Eu vejo um erro: git: 'remote-codecommit' não é um comando git](#)
- [Eu vejo um erro: fatal: Não foi possível encontrar o assistente remoto para 'codecommit'](#)
- [Erro de clonagem: não consigo clonar um CodeCommit repositório de um IDE](#)
- [Erro de push ou pull: não consigo enviar ou extrair commits de um IDE para um repositório CodeCommit](#)
- [Erro de instalação: vejo um externally-managed-environment erro quando tento instalar git-remote-codecommit](#)

Eu vejo um erro: git: 'remote-codecommit' não é um comando git

Problema: Ao tentar usar git-remote-codecommit, você vê um erro que não git-remote-codecommit é um comando git. Veja 'git --help'.

Possíveis correções: o motivo mais comum para esse erro é que você não adicionou o git-remote-codecommit executável ao PATH ou que a string contém um erro de sintaxe. Isso pode acontecer quando falta um hífen entre git e remote-codecommit, ou quando um git extra é colocado antes. git-remote-codecommit

Você também pode encontrar esse erro se tiver atualizado sua instalação local para uma versão sem suporte. Para obter mais informações sobre as versões compatíveis do Python, consulte. [git-remote-codecommit](#)

Para obter mais informações sobre como configurar e usar git-remote-codecommit, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Eu vejo um erro: fatal: Não foi possível encontrar o assistente remoto para 'codecommit'

Problema: Ao tentar usar git-remote-codecommit, você vê um erro informando “fatal: Não foi possível encontrar o ajudante remoto para 'codecommit'”.

Correções possíveis: os motivos mais comuns para essa mensagem de erro são:

- A configuração não está completa para git-remote-codecommit
- Você instalou git-remote-codecommit em um local que não está em seu caminho ou não está configurado como parte da variável de Path ambiente
- Python não está no seu caminho ou não está configurado como parte da variável de ambiente Path
- Você está usando um terminal ou uma janela de linha de comando que não foi reiniciada desde a conclusão da git-remote-codecommit instalação

Para obter mais informações sobre como configurar e usar git-remote-codecommit, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Erro de clonagem: não consigo clonar um CodeCommit repositório de um IDE

Problema: Ao tentar clonar um CodeCommit repositório em um IDE, você vê um erro que diz que o endpoint ou URL não é válido.

Possíveis correções: nem todas IDEs oferecem suporte ao URL usado git-remote-codecommit durante a clonagem. Clone o repositório localmente a partir do terminal ou da linha de comando e, depois, adicione esse repositório local ao IDE. Para obter mais informações, consulte [Etapa 3: Conectar-se ao CodeCommit console e clonar o repositório](#).

Erro de push ou pull: não consigo enviar ou extrair commits de um IDE para um repositório CodeCommit

Problema: ao tentar extrair ou enviar código de um IDE, você vê um erro de conexão.

Possíveis correções: a razão mais comum para esse erro é que o IDE não é compatível com assistentes remotos do Git, como git-remote-codecommit. Em vez de usar a funcionalidade do IDE

para confirmar, enviar e extrair código, atualize o repositório local manualmente a partir da linha de comando ou do terminal usando comandos Git.

Para obter mais informações sobre assistentes remotos e Git, consulte a [documentação do Git](#).

Erro de instalação: vejo um externally-managed-environment erro quando tento instalar git-remote-codecommit

Problema: quando tenta executar o comando `pip install git-remote-codecommit`, você vê um erro indicando que o ambiente é gerenciado externamente.

Possíveis correções: o motivo mais comum para esse erro é que você está executando uma distribuição (distro) do Python que define um arquivo de marcador EXTERNALLY-MANAGED. A melhor solução para isso é criar e usar um ambiente virtual.

Para obter mais informações sobre ambientes gerenciados externamente e Python, consulte [Ambientes gerenciados externamente](#) e [Instalar pacotes em um ambiente virtual usando pip e venv](#) na documentação do Python.

Solução de problemas de conexões SSH com AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas comuns ao usar o SSH para se conectar aos CodeCommit repositórios.

Tópicos

- [Erro de acesso: o upload da chave pública no IAM foi feito com sucesso, mas a conexão falha nos sistemas Linux, macOS ou Unix](#)
- [Erro de acesso: o upload da chave pública no IAM foi feito com sucesso e o SSH foi testado com sucesso, mas a conexão falha em sistemas Windows](#)
- [Desafio de autenticação: a autenticidade do host não pode ser estabelecida ao se conectar a um repositório CodeCommit](#)
- [Erro do IAM: "Formato inválido" ao tentar adicionar uma chave pública ao IAM](#)
- [Preciso acessar CodeCommit repositórios em várias contas da Amazon Web Services com credenciais SSH](#)
- [Git no Windows: o emulador de Bash ou a linha de comando congelam durante a tentativa de se conectar usando SSH](#)

- [O formato de chave pública requer especificação em algumas distribuições do Linux](#)
- [Erro de acesso: chave pública SSH negada ao se conectar a um repositório CodeCommit](#)

Erro de acesso: o upload da chave pública no IAM foi feito com sucesso, mas a conexão falha nos sistemas Linux, macOS ou Unix

Problema: quando você tenta se conectar a um endpoint SSH para se comunicar com um CodeCommit repositório, seja ao testar a conexão ou clonar um repositório, a conexão falha ou é recusada.

Possíveis correções: o ID da chave SSH atribuído para a chave pública no IAM pode não estar associado à sua tentativa de conexão. [Você pode não ter configurado um arquivo de configuração](#), pode não ter acesso ao arquivo de configuração, outra configuração pode estar impedindo a leitura bem-sucedida do arquivo de configuração, você pode ter fornecido o ID da chave errado, ou pode ter fornecido o ID do usuário do IAM em vez do ID da chave.

O ID da chave SSH pode ser encontrado no console do IAM, no perfil do usuário do 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 você tiver mais de uma chave SSH IDs carregada, as chaves serão listadas alfabeticamente por ID da chave, não pela data de upload. Verifique se você copiou o ID da chave que está associado à data de upload correta.

Experimente testar a conexão com o seguinte comando:

```
ssh Your-SSH-Key-ID@git-codecommit.us-east-2.amazonaws.com
```

Se você vir uma mensagem de êxito após confirmar a conexão, o ID da chave SSH será válido. Edite seu arquivo de configuração para associar suas tentativas de conexão com sua chave pública no IAM. Se não quiser editar o arquivo de configuração, você poderá prefaciá-las todas as tentativas

de conexão ao repositório com o ID da chave SSH. Por exemplo, se você quisesse clonar um repositório chamado *MyDemoRepo* sem modificar seu arquivo de configuração para associar suas tentativas de conexão, execute o seguinte comando:

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

Para obter mais informações, consulte [Para conexões SSH no Linux, macOS ou Unix](#).

Erro de acesso: o upload da chave pública no IAM foi feito com sucesso e o SSH foi testado com sucesso, mas a conexão falha em sistemas Windows

Problema: quando você tenta usar um endpoint SSH para clonar ou se comunicar com um CodeCommit repositório, aparece uma mensagem de erro contendo a frase. No supported authentication methods available

Correções possíveis: o motivo o mais comum para esse erro é que você tem uma definição de variável de ambiente do sistema Windows que direciona o Windows a usar outro programa quando você tenta usar o SSH. Por exemplo, você pode ter definido uma variável GIT_SSH para apontar para um dos conjunto de ferramentas PuTTY (plink.exe). Essa pode ser uma configuração legada ou pode ser necessária para outros programas instalados em seu computador. Se tiver certeza de que essa variável de ambiente não é necessária, você poderá removê-la abrindo as propriedades do sistema e excluindo a variável de ambiente.

Para resolver esse problema, abra um emulador de Bash e teste sua conexão SSH novamente, mas inclua GIT_SSH_COMMAND="SSH" como prefixo. Por exemplo, para clonar um repositório usando SSH:

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

Um problema semelhante pode ocorrer se sua versão do Windows exigir que você inclua o ID da chave SSH como parte da cadeia de conexão ao se conectar por SSH na linha de comando do Windows. Teste sua conexão novamente, dessa vez incluindo o ID da chave SSH copiado do IAM como parte do comando. Por exemplo:

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

Desafio de autenticação: a autenticidade do host não pode ser estabelecida ao se conectar a um repositório CodeCommit

Problema: quando você tenta usar um endpoint SSH para se comunicar com um CodeCommit repositório, aparece uma mensagem de aviso contendo a frase The authenticity of host '*host-name*' can't be established.

Correções possíveis: suas credenciais podem não ter sido configuradas adequadamente. Siga as instruções em [Para conexões SSH no Linux, macOS ou Unix](#) ou [Para conexões SSH no Windows](#).

Se você seguiu essas etapas e o problema persistir, alguém pode estar tentando um man-in-the-middle ataque. Quando aparecer a seguinte mensagem, digite no e pressione Enter.

```
Are you sure you want to continue connecting (yes/no)?
```

Certifique-se de que a impressão digital e a chave pública das CodeCommit conexões correspondam às documentadas nos tópicos de configuração do SSH antes de continuar com a conexão.

Impressões digitais públicas para CodeCommit

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.us-east-2.amazonaws.com	MD5	a9:6d:03:ed:08:42: 21:be:06:e1:e0:2a: d1:75:31:5e
git-codecommit.us-east-2.amazonaws.com	SHA256	3lB1W2g5xn/NA2Ck6d yeJIrQ0Wvn7n8UEs56 fG6ZIZQ
git-codecommit.us-east-1.amazonaws.com	MD5	a6:9c:7d:bc:35:f5: d4:5f:8b:ba:6f:c8: bc:d4:83:84
git-codecommit.us-east-1.amazonaws.com	SHA256	eLMY1j0DKA4uvDZc1/ KgtIayZANwX6t8+8is PtotBoY

Servidor	Tipo de hash de criptografia	Impressão digital
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/f0iq13R1vqg 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
git-codecommit.ap-southeast-2.amazonaws.com	MD5	7b:d2:c1:24:e6:91: a5:7b:fa:c1:0c:35: 95:87:da:a0

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.ap-southeast-2.amazonaws.com	SHA256	nYp+gHas80HY3DqbP4yanCDFhqDVjseefVbH 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	grceUDWubo4MzG1NoaKZKUfrgPvfN3ijli0n Qr11TZA
git-codecommit.eu-central-1.amazonaws.com	MD5	74:5a:e8:02:fc:b2: 9c:06:10:b4:78:84: 65:94:22:2d
git-codecommit.eu-central-1.amazonaws.com	SHA256	MwGrkiEki8QkkBt1AgXbYt0hoZYBnZF62VY5 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	eegAPQrWY9YsYo9ZHIK0mxfXbHZAzd8Eya 53Qcwko

Servidor	Tipo de hash de criptografia	Impressão digital
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 BODLpzg
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
git-codecommit.ap-south-2.amazonaws.com	MD5	bc:cc:9f:15:f8:f3: 58:a2:68:65:21:e2: 23:71:8d:ce

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.ap-south-2.amazonaws.com	SHA256	Xe0CyZE0vgR5Xa2YUG qf+jn8/Ut7l7nX/Cms lSFNEig
git-codecommit.ca-central-1.amazonaws.com	MD5	9f:7c:a2:2f:8c:b5: 74:fd:ab:b7:e1:fd: af:46:ed:23
git-codecommit.ca-central-1.amazonaws.com	SHA256	Qz5puafQdANVprLlj6 r0Qyh4lCNsF6ob61dG cPtFS7w
git-codecommit.eu-west-3.amazonaws.com	MD5	1b:7f:97:dd:d7:76: 8a:32:2c:bd:2c:7b: 33:74:6a:76
git-codecommit.eu-west-3.amazonaws.com	SHA256	uw7c2FL564jVoFgtc+ ikzILnKBsZz7t9+CFd SJjKbLI
git-codecommit.us-gov-west-1.amazonaws.com	MD5	9f:6c:19:3b:88:cd: e8:88:1b:9c:98:6a: 95:31:8a:69
git-codecommit.us-gov-west-1.amazonaws.com	SHA256	djXQoSIFcg8vHe0KVH 1xW/g0F9X37tWTqu4H kng75x4
git-codecommit.us-gov-east-1.amazonaws.com	MD5	00:8d:b5:55:6f:05: 78:05:ed:ea:cb:3f: e6:f0:62:f2
git-codecommit.us-gov-east-1.amazonaws.com	SHA256	fVb+R0z7qW7minenW+ rUpAABRCRBTCzmETAJ EQrg98

Servidor	Tipo de hash de criptografia	Impressão digital
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+NToCGgjRHeKiBu01 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
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

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.cn-northwest-1.amazonaws.com.cn	SHA256	wqjd6eHd0+m0Bx+dCNuL0omUoCNjaDtZiEpWj5TmCfQ
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	1yXrWbCg3uQmJr11XxB/ASR7ugW1Ysf5yzY0JbudHsI
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	2B815B9F0AvwLnRxSVxUz4kDYmtEQUGGdQYP80QLXhA
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/cnsDZjUpyNGXdE5pjHYimqJrQZ31eTgqJHM
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	uFxhp51kUWh1eTLeYbxQVYm4RnNLNZ5Dbdm1cgcS1/8

Erro do IAM: "Formato inválido" ao tentar adicionar uma chave pública ao IAM

Problema: no IAM, ao tentar configurar para usar SSH CodeCommit, aparece uma mensagem de erro contendo a frase `Invalid format` quando você tenta adicionar sua chave pública.

Possíveis correções: o IAM requer que a chave pública seja codificada no formato `ssh-rsa` ou PEM. Ele aceita chaves públicas somente no formato OpenSSH e tem requisitos adicionais, conforme especificado [em Usar chaves SSH no Guia do usuário do CodeCommit IAM](#). Se fornecer sua chave pública em outro formato ou se ela não tiver o número necessário de bits, você verá esse erro.

- Quando você copiou a chave pública SSH, o sistema operacional pode ter introduzido quebras de linha. Verifique se não há quebras de linha na chave pública adicionada ao IAM.
- Alguns sistemas operacionais Windows não usam o formato OpenSSH. Para gerar um par de chaves e copiar o formato OpenSSH exigido pelo IAM, consulte [the section called “Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit”](#).

Para obter mais informações sobre os requisitos das chaves SSH no IAM, consulte [Usar chaves SSH CodeCommit no Guia](#) do usuário do IAM.

Preciso acessar CodeCommit repositórios em várias contas da Amazon Web Services com credenciais SSH

Problema: quero configurar o acesso SSH aos CodeCommit repositórios em mais de uma conta da Amazon Web Services.

Possíveis correções: você pode criar pares de public/private chaves SSH exclusivos para cada conta da Amazon Web Services e configurar o IAM com cada chave. Em seguida, você pode configurar seu arquivo `~/.ssh/config` com informações sobre cada ID de usuário do IAM associado à chave pública. Por exemplo:

```
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.
```

Nessa configuração, você poderá substituir 'git-codecommit.us-east-1.amazonaws.com' por 'codecommit-2'. Por exemplo, para clonar um repositório em sua segunda conta da Amazon Web Services:

```
git clone ssh://codecommit-2/v1/repos/YourRepositoryName
```

Para configurar um remoto para seu repositório, execute git remote add. Por exemplo:

```
git remote add origin ssh://codecommit-2/v1/repos/YourRepositoryName
```

Para obter mais exemplos, consulte [esta postagem no fórum](#) e [esta contribuição em GitHub](#).

Git no Windows: o emulador de Bash ou a linha de comando congelam durante a tentativa de se conectar usando SSH

Problema: depois de configurar o acesso SSH para Windows e confirmar a conectividade na linha de comando ou no terminal, aparece uma mensagem de que a chave host do servidor não está em cache no registro e a solicitação para armazenar a chave em cache está congelada (não aceita a entrada y/n/return) quando você tenta usar comandos como git pull, git push ou git clone no prompt de comando ou no emulador de Bash.

Correções possíveis: a causa mais comum para esse erro é que o ambiente do Git está configurado para usar algo que não o OpenSSH para autenticação (provavelmente o PuTTY). É sabido que isso causa problemas com o armazenamento em cache de chaves em algumas configurações. Por corrigir esse problema, tente o seguinte:

- Abra um emulador de Bash e adicione o parâmetro GIT_SSH_COMMAND="ssh" antes do comando do Git. Por exemplo, se você está tentando enviar um repositório por push, em vez de digitar git push, digite:

```
GIT_SSH_COMMAND="ssh" git push
```

- Se você tiver o PuTTY instalado, abra o PuTTY e, em Nome do host (ou endereço IP), insira o CodeCommit endpoint que você deseja alcançar (por exemplo, git-codecommit.us-east-2.amazonaws.com). Escolha Open (Abrir). Quando solicitado pelo alerta de segurança do PuTTY, selecione Yes (Sim) para armazenar a chave em cache permanentemente.
- Renomeie ou exclua a variável de ambiente GIT_SSH se não a estiver mais utilizando. Em seguida, abra um novo prompt de comando ou uma sessão do emulador de Bash e teste o comando novamente.

Para outras soluções, consulte [Git congelando clone/pull continuamente na chave de armazenamento no cache no Stack Overflow](#).

O formato de chave pública requer especificação em algumas distribuições do Linux

Problema: ao tentar configurar um par de chaves público-privado, você recebe um erro.

Possíveis correções: algumas distribuições do Linux exigem uma linha adicional de configuração no arquivo `~/.ssh/config` que especifica os tipos aceitos de chaves públicas. Para obter mais informações, consulte a documentação para sua distribuição sobre `PubkeyAcceptedKeyTypes`.

Erro de acesso: chave pública SSH negada ao se conectar a um repositório CodeCommit

Problema: Quando você tenta usar um endpoint SSH para se comunicar com um CodeCommit repositório, aparece uma mensagem de erro contendo a frase. `Error: public key denied`

Correções possíveis: o motivo mais comum para esse erro é que você não concluiu a configuração para conexões SSH. Configure um par de chaves SSH pública e privada e, em seguida, associe a chave pública ao seu usuário do IAM. Para obter mais informações sobre a configuração de SSH, consulte [Para conexões SSH no Linux, macOS ou Unix](#) e [Para conexões SSH no Windows](#).

Solução de problemas do auxiliar de credenciais e das conexões HTTPS com AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas comuns ao usar o auxiliar de credenciais incluído com o AWS CLI e HTTPS para se conectar aos repositórios. CodeCommit

 Note

Embora o auxiliar de credenciais seja um método compatível para se conectar CodeCommit usando acesso federado, um provedor de identidade ou credenciais temporárias, o método recomendado é instalar e usar o utilitário. `git-remote-codecommit` Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

Tópicos

- [Eu recebo um erro ao executar o comando `git config` para configurar o assistente de credenciais](#)
- [Vejo um erro ao ler o nome do usuário quando tento clonar um repositório](#)
- [Recebo um erro de comando não encontrado no Windows ao usar o assistente de credenciais](#)
- [Sou solicitado a fornecer um nome de usuário quando me conecto a um repositório CodeCommit](#)
- [Git para macOS: eu configurei o assistente de credenciais com êxito, mas agora o acesso ao meu repositório é negado \(403\)](#)
- [Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado \(403\)](#)

Eu recebo um erro ao executar o comando **git config** para configurar o assistente de credenciais

Problema: ao tentar executar o comando `git config` para configurar o auxiliar de credenciais para se comunicar com um CodeCommit repositório, você vê um erro informando que há poucos argumentos ou um prompt de uso sugerindo comandos e sintaxe do `Git config`.

Possíveis correções: o motivo mais comum para esse erro é que aspas simples são usadas para o comando em um sistema operacional Windows ou aspas duplas são usadas para o comando em um sistema operacional Linux, macOS ou Unix. A sintaxe correta é a seguinte:

- Windows: `git config --global credential.helper "!aws codecommit credential-helper %@"`
- Linux, macOS ou Unix: `git config --global credential.helper '!aws codecommit credential-helper %@'`

Vejo um erro ao ler o nome do usuário quando tento clonar um repositório

Problema: quando tenta clonar um repositório usando o assistente de credenciais, você vê um erro informando que o sistema não conseguiu ler o nome de usuário da URL do repositório.

Correções possíveis: o motivo mais comum para esse erro é que o arquivo `.gitconfig` não foi criado ou configurado corretamente no seu computador. Abra seu arquivo `.gitconfig` e verifique se o assistente de credenciais está configurado corretamente. Se você estiver em um computador executando Linux, macOS ou Unix, verifique também se o valor de `$HOME` está definido corretamente para o seu sistema.

Recebo um erro de comando não encontrado no Windows ao usar o assistente de credenciais

Problema: Depois de atualizar a AWS CLI, as conexões auxiliares de credenciais aos CodeCommit repositórios falham com `aws codecommit credential-helper $@ get: aws: command not found`

Causa: O motivo mais comum para esse erro é que sua versão da AWS CLI foi atualizada para uma versão que usa Python 3. Há um problema conhecido com o pacote MSI. Para verificar se você tem uma das versões afetadas, abra uma linha de comando e execute o seguinte comando: `aws --version`

Se a saída da versão do Python começar com 3, você terá uma versão afetada. Por exemplo:

```
aws-cli/1.16.62 Python/3.6.2 Darwin/16.7.0 botocore/1.12.52
```

Correções possíveis: Para contornar este problema, siga uma das seguintes opções:

- Instale e configure o AWS CLI no Windows usando Python e pip em vez do MSI. Para obter mais informações, consulte [Instalar Python, pip e the no AWS CLI Windows](#).
- Edite manualmente o arquivo `.gitconfig` para alterar a seção `[credential]` para apontar explicitamente para `aws.cmd` em seu computador local. Por exemplo:

```
[credential]
  helper = !"C:\\Program Files\\Amazon\\AWSCLI\\bin\\aws.cmd\" codecommit
credential-helper $@
  UseHttpPath = true
```

- Execute o comando `git config` para atualizar o arquivo `.gitconfig` para fazer referência explicitamente a `aws.cmd` e atualize manualmente sua variável de ambiente `PATH` para incluir o caminho do comando, conforme necessário. Por exemplo:

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

Sou solicitado a fornecer um nome de usuário quando me conecto a um repositório CodeCommit

Problema: Quando você tenta usar o auxiliar de credenciais para se comunicar com um CodeCommit repositório, aparece uma mensagem solicitando seu nome de usuário.

Possíveis correções: configure seu AWS perfil ou verifique se o perfil que você está usando é aquele com o qual você configurou para trabalhar CodeCommit. Para obter mais informações sobre a configuração, consulte [Etapas de configuração para conexões HTTPS com AWS CodeCommit repositórios no Linux, macOS ou Unix com o auxiliar de credenciais AWS CLI](#) ou [Etapas de configuração para conexões HTTPS com AWS CodeCommit repositórios no Windows com o auxiliar de AWS CLI credenciais](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Gerenciamento de chaves de acesso para usuários do IAM](#) e [Como consigo credenciais?](#)

Git para macOS: eu configurei o assistente de credenciais com êxito, mas agora o acesso ao meu repositório é negado (403)

Problema: no macOS, o assistente de credenciais parece não acessar ou usar suas credenciais como esperado. Isso pode ser causado por dois problemas diferentes:

- O AWS CLI está configurado para um Região da AWS diferente daquele em que o repositório existe.
- O utilitário Keychain Access salvou credenciais já expiradas.

Possíveis correções: Para verificar se o AWS CLI está configurado para a região correta, execute o `aws configure` comando e revise as informações exibidas. Se o CodeCommit repositório estiver em um Região da AWS local diferente do mostrado para o AWS CLI, você deverá executar o `aws`

configure comando e alterar os valores para aqueles apropriados para aquela região. Para obter mais informações, consulte [Etapa 1: Configuração inicial para CodeCommit](#).

A versão padrão do Git lançada para OS X e macOS usa o utilitário Acesso às Chaves para salvar as credenciais geradas. Por motivos de segurança, a senha gerada para acessar seu CodeCommit repositório é temporária, portanto, as credenciais armazenadas no chaveiro param de funcionar após cerca de 15 minutos. Se você estiver acessando o Git somente com CodeCommit, tente o seguinte:

1. No terminal, execute o comando `git config` para localizar o arquivo de configuração do Git (`gitconfig`) em que o utilitário Keychain Access está definido. Dependendo do seu sistema local e de suas preferências, você pode ter mais de um arquivo `gitconfig`.

```
git config -l --show-origin | grep credential
```

Na saída desse comando, procure resultados semelhantes a:

```
file://path/to/gitconfig credential.helper=osxkeychain
```

O arquivo listado no início dessa linha é o arquivo de configuração do Git que você deve editar.

2. Para editar o arquivo de configuração do Git, use um editor de texto simples e execute o seguinte comando:

```
nano /usr/local/git/etc/gitconfig
```

3. Modifique a configuração usando uma das seguintes estratégias:

- Comente ou exclua a seção de credenciais que contém `helper = osxkeychain`. Por exemplo:

```
# helper = osxkeychain
```

- Atualize as seções assistente de credenciais `aws credential helper` e `osxkeychain` para ter contexto. Por exemplo, se `osxkeychain` for usado para se autenticar GitHub em:

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

Nessa configuração, o Git usará o assistente `osxkeychain` quando o host remoto corresponder a `"https://github.com"` e ao assistente de credenciais quando o host remoto corresponder a `"https://git-codecommit\us-east-1\amazonaws.com"`.

- Inclua um assistente de string vazio antes do assistente de credenciais. Por exemplo, para não usar o `osxkeychain` auxiliar ao usar o perfil nomeado *CodeCommitProfile* com a CLI:

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

 Tip

Você também pode configurar a linha após a linha auxiliar de string vazia para não corresponder CodeCommit se quiser excluir todos os perfis:

```
helper = !aws codecommit credential-helper $@
```

Como alternativa, se você deseja continuar a usar o utilitário Keychain Access para armazenar em cache as credenciais para outros repositórios Git, modifique o cabeçalho em vez de comentar a linha. Por exemplo, para permitir credenciais em cache GitHub, você pode modificar o cabeçalho da seguinte forma:

```
[credential "https://github.com"]
  helper = osxkeychain
```

Se você estiver acessando outros repositórios com o Git, você pode configurar o utilitário Keychain Access para que ele não forneça credenciais para seus repositórios. CodeCommit Como configurar o utilitário Acesso às Chaves:

1. Abra o utilitário Acesso às Chaves. (Você pode usar o Finder para localizá-lo.)
2. Pesquise `git-codecommit.us-east-2.amazonaws.com` e *us-east-2* substitua pelo Região da AWS local onde o repositório existe. Destaque a linha, abra o menu de contexto (clique com o botão direito do mouse) e escolha Obter Informações.

3. Escolha a guia Controle de Acesso.
4. Em Confirmar antes de permitir o acesso, escolha `git-credential-osxkeychain` e, em seguida, selecione o sinal de menos para removê-la da lista.

Note

Após remover `git-credential-osxkeychain` da lista, você verá uma caixa de diálogo sempre que executar um comando do Git. Escolha Negar para continuar. Se você achar os pop-ups um incômodo, aqui estão algumas alternativas:

- Conecte-se CodeCommit usando credenciais SSH ou Git em vez do auxiliar de credenciais com HTTPS. Para obter mais informações, consulte [Para conexões SSH no Linux, macOS ou Unix](#) e [Configuração para usuários de HTTPS usando credenciais do Git](#).
- No utilitário Acesso às Chaves, na guia Controle de Acesso, para `git-codecommit.us-east-2.amazonaws.com`, escolha a opção Permitir que todos os aplicativos acessem este item (o acesso a esse item não é restrito). Isso impedirá os pop-ups, mas as credenciais expirarão eventualmente (em média, leva cerca de 15 minutos) e você verá uma mensagem de erro 403. Quando isso ocorrer, você deverá excluir o item do conjunto de chaves para restaurar a funcionalidade.
- Instalar uma versão do Git que não utilize o conjunto de chaves por padrão; ou
- Considere uma solução de scripts para excluir o item do conjunto de chaves. Para visualizar uma amostra gerada pela comunidade de uma solução por scripts, consulte [Mac OS X Script to Periodically Delete Cached Credentials in the OS X Certificate Store](#) em [Integrações de produtos e serviços](#).

Se quiser que o Git pare de usar o utilitário Keychain Access totalmente, você poderá configurar o Git para parar de usar `osxkeychain` como o assistente de credenciais. Por exemplo, se você abrir um terminal e executar o comando `git config --system credential.helper` e ele retornar `osxkeychain`, o Git estará configurado para usar o utilitário Keychain Access. Você poderá alterar isso executando o seguinte comando:

```
git config --system --unset credential.helper
```

Lembre-se de que executar esse comando com a `--system` opção altera o comportamento do Git em todo o sistema para todos os usuários, e isso pode ter consequências não intencionais para

outros usuários ou para outros repositórios se você estiver usando outros serviços de repositório além do. CodeCommit Lembre-se também de que essa abordagem pode exigir o uso de `sudo` e que sua conta pode não ter permissões suficientes do sistema para aplicar essa alteração. Verifique se o comando foi aplicado com êxito executando o comando `git config --system credential.helper` novamente. Para obter mais informações, consulte [Personalização do Git – Configuração do Git](#) e [este artigo no Stack Overflow](#).

Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado (403)

Problema: no Windows, o assistente de credenciais parece não acessar ou usar suas credenciais como esperado. Isso pode ser causado por diferentes problemas:

- O AWS CLI está configurado para um Região da AWS diferente daquele em que o repositório existe.
- Por padrão, o Git para Windows instala um utilitário Git Credential Manager que não é compatível com CodeCommit conexões que usam o auxiliar de credenciais. AWS Quando instalado, ele faz com que as conexões com o repositório falhem, mesmo que o auxiliar de credenciais tenha sido instalado com o AWS CLI e configurado para conexões com o. CodeCommit
- Algumas versões do Git para Windows podem não ser totalmente compatíveis com a [RFC 2617](#) e a [RFC 4559](#), o que pode causar problemas com as credenciais do Git e o assistente de credenciais incluído com a AWS CLI. Para obter mais informações, consulte [Version 2.11.0\(3\) does not ask for username/password](#).

Correções possíveis:

- Se você estiver tentando usar o auxiliar de credenciais incluído no AWS CLI, considere conectar-se com as credenciais do Git por HTTPS em vez de usar o auxiliar de credenciais. As credenciais do Git configuradas para seu usuário do IAM são compatíveis com o Git Credential Manager para Windows, diferente do assistente de credenciais do AWS CodeCommit. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#).

Se você quiser usar o auxiliar de credenciais, para verificar se ele AWS CLI está configurado corretamente Região da AWS, execute o `aws configure` comando e revise as informações exibidas. Se o CodeCommit repositório estiver em um Região da AWS local diferente do mostrado para o AWS CLI, você deverá executar o `aws configure` comando e alterar os valores para aqueles

apropriados para aquela região. Para obter mais informações, consulte [Etapa 1: Configuração inicial para CodeCommit](#).

- Se possível, desinstale e reinstale o Git para Windows. Ao instalar o Git para Windows, desmarque a caixa de seleção com a opção de instalar o utilitário Git Credential Manager. Esse gerenciador de credenciais não é compatível com o assistente de credenciais do AWS CodeCommit. Se você instalou o Git Credential Manager ou outro utilitário de gerenciamento de credenciais e não quer desinstalá-lo, pode modificar o arquivo `.gitconfig` e adicionar gerenciamentos de credenciais para o CodeCommit:
1. Abra o Painel de Controle, escolha Gerenciador de Credenciais e remova todas as credenciais armazenadas para. CodeCommit
 2. Abra o arquivo `.gitconfig` em um editor de texto simples, como o Bloco de Notas.

 Note

Se você trabalha com vários perfis do Git, talvez tenha arquivos `.gitconfig` locais e globais. Certifique-se de editar o arquivo apropriado.

3. Adicione a seguinte seção ao arquivo `.gitconfig`:

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

4. Salve o arquivo e abra uma nova sessão de linha de comando antes de tentar se conectar novamente.

Você também pode usar essa abordagem se quiser usar o auxiliar de credenciais para AWS CodeCommit se conectar a CodeCommit repositórios e outro sistema de gerenciamento de credenciais ao se conectar a outros repositórios hospedados, como repositórios. GitHub

Para restaurar o assistente de credenciais usado como padrão, você pode usar a opção `--system` em vez de `--global` ou `--local` ao executar o comando `git config`.

- Se você está usando credenciais do Git em um computador Windows, pode tentar solucionar quaisquer problemas de não conformidade do RFC incluindo seu nome de usuário das credenciais do Git como parte da cadeia de conexão. Por exemplo, para contornar o problema e clonar um repositório chamado *MyDemoRepo* na região Leste dos EUA (Ohio):


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

Note

Essa abordagem não funcionará se você tiver um caractere @ no seu nome de usuário das credenciais do Git. Você deve primeiro codificar o caractere em URL (processo também conhecido como [codificação percentual](#)).

Solução de problemas de clientes Git e AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas comuns ao usar o Git com repositórios do AWS CodeCommit. Para solucionar problemas relacionados a clientes do Git ao usar HTTPS ou SSH, consulte também [Solução de problemas com credenciais do Git \(HTTPS\)](#), [Solução de problemas com conexões SSH](#) e [Solução de problemas com o assistente de credenciais \(HTTPS\)](#).

Tópicos

- [Erro do Git: Error: RPC failed; result=56, HTTP code = 200 fatal: The remote end hung up unexpectedly](#)
- [Erro do Git: comandos de atualização de referência em demasia](#)
- [Erro do Git: o envio por HTTPS está quebrado em algumas versões do Git](#)
- [Erro do Git: "gnutls_handshake\(\) failed"](#)
- [Erro do Git: o Git não consegue encontrar o CodeCommit repositório ou não tem permissão para acessar o repositório](#)
- [Git no Windows: No supported authentication methods available \(publickey\)](#)

Erro do Git: Error: RPC failed; result=56, HTTP code = 200 fatal: The remote end hung up unexpectedly

Problema: ao enviar por push uma grande alteração, um grande número de alterações ou um grande repositório, as conexões HTTPS de execução prolongada costumam ser encerradas prematuramente devido a problemas de rede ou configurações do firewall.

Correções possíveis: em vez disso, envie com SSH por push ou, ao migrar um grande repositório, siga as etapas em [Migrar um repositório em incrementos](#). Além disso, verifique se você não está excedendo os limites de tamanho para arquivos individuais. Para obter mais informações, consulte [Cotas](#).

Erro do Git: comandos de atualização de referência em demasia

Problema: o número máximo de atualizações de referência por push é 4.000. Esse erro aparece quando o push contém mais de 4.000 atualizações de referência.

Correções possíveis: teste enviar por push ramificações e tags individualmente com `git push --all` e `git push --tags`. Se houver tags demais, divida-as em vários pushes. Para obter mais informações, consulte [Cotas](#).

Erro do Git: o envio por HTTPS está quebrado em algumas versões do Git

Problema: um problema com a atualização do curl para 7.41.0 faz com que a autenticação Digest baseada em SSPI falhe. As versões do Git sabidamente afetadas incluem 1.9.5.msysgit.1. Algumas versões do Git para Windows podem não ser totalmente compatíveis com a [RFC 2617](#) e a [RFC 4559](#), o que pode causar problemas com as conexões HTTPS que usam credenciais do Git ou o assistente de credenciais incluído na AWS CLI.

Correções possíveis: verifique os problemas conhecidos em sua versão do Git ou use uma versão anterior ou posterior. Para obter mais informações sobre msysgit, consulte [Push to HTTPS Is Broken nos fóruns](#). GitHub Para obter mais informações sobre problemas de versão do Git para Windows, consulte o tópico [A versão 2.11.0\(3\) não solicita nome de usuário/senha](#).

Erro do Git: "gnutls_handshake() failed"

Problema: no Linux, quando você tenta usar o Git para se comunicar com um CodeCommit repositório, aparece uma mensagem de erro contendo a frase. `error: gnutls_handshake() failed`

Correções possíveis: Compile o Git com o OpenSSL. Para conhecer uma abordagem, consulte ["Error: gnutls_handshake\(\) failed" When Connecting to HTTPS Servers](#) no fórum do Ask Ubuntu.

Como alternativa, use SSH em vez de HTTPS para se comunicar com os CodeCommit repositórios.

Erro do Git: o Git não consegue encontrar o CodeCommit repositório ou não tem permissão para acessar o repositório

Problema: uma barra final na cadeia de conexão pode fazer com que as tentativas de conexão falhem.

Correções possíveis: certifique-se de que você forneceu o nome e a cadeia de conexão corretos para o repositório e que não há barras finais. Para obter mais informações, consulte [Conexão a um repositório](#).

Git no Windows: No supported authentication methods available (publickey)

Problema: depois de configurar o acesso SSH para Windows, você vê um erro de acesso recusado ao tentar usar comandos como git pull, git push ou git clone.

Correções possíveis: a causa mais comum para esse erro é que há uma variável de ambiente GIT_SSH em seu computador e ela é configurada para oferecer suporte a outro utilitário de conexão, como PuTTY. Por corrigir esse problema, tente o seguinte:

- Abra um emulador de Bash e adicione o parâmetro GIT_SSH_COMMAND="ssh" antes do comando do Git. Por exemplo, se você está tentando clonar um repositório, em vez de executar git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo, execute:

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

- Renomeie ou exclua a variável de ambiente GIT_SSH se não a estiver mais utilizando. Em seguida, abra um novo prompt de comando ou uma sessão do emulador de Bash e teste o comando novamente.

Para obter mais informações sobre como solucionar problemas com o Git no Windows ao usar o SSH, consulte [Solução de problemas com conexões SSH](#).

Solução de problemas de erros de acesso e AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar erros de acesso ao se conectar a repositórios do AWS CodeCommit .

Tópicos

- [Erro de acesso: sou solicitado a fornecer um nome de usuário e uma senha quando me conecto a um CodeCommit repositório a partir do Windows](#)
- [Erro de acesso: chave pública negada ao se conectar a um CodeCommit repositório](#)
- [Erro de acesso: mensagem “Taxa excedida” ou “429” ao se conectar a um repositório CodeCommit](#)

Erro de acesso: sou solicitado a fornecer um nome de usuário e uma senha quando me conecto a um CodeCommit repositório a partir do Windows

Problema: Quando você tenta usar o Git para se comunicar com um CodeCommit repositório, você vê uma caixa de diálogo que solicita seu nome de usuário e senha.

Correções possíveis: isso pode ser o sistema de gerenciamento de credenciais interno do Windows. Dependendo da sua configuração, siga um destes procedimentos:

- Se você estiver usando HTTPS com credenciais do Git, suas credenciais do Git ainda não estão armazenadas no sistema. Forneça as credenciais do Git e continue. Você não deve ser avisado novamente. Para obter mais informações, consulte [Para usuários de HTTPS usando credenciais do Git](#).
- Se você estiver usando HTTPS com o auxiliar de credenciais AWS CodeCommit, ele não é compatível com o sistema de gerenciamento de credenciais do Windows. Escolha Cancelar.

Isso também pode indicar que você instalou o Git Credential Manager ao instalar o Git para Windows. O Git Credential Manager não é compatível com o auxiliar de credenciais incluído no CodeCommit AWS CLI. Considere desinstalar o Git Credential Manager. Você também pode instalar e configurar git-remote-codecommit como uma alternativa ao uso do assistente de credenciais para o CodeCommit.

Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#), [Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais](#) e [Git para Windows: eu instalei o Git para Windows, mas o acesso ao meu repositório é negado \(403\)](#).

Erro de acesso: chave pública negada ao se conectar a um CodeCommit repositório

Problema: Quando você tenta usar um endpoint SSH para se comunicar com um CodeCommit repositório, aparece uma mensagem de erro contendo a frase. `Error: public key denied`

Correções possíveis: o motivo mais comum para esse erro é que você não concluiu a configuração para conexões SSH. Configure um par de chaves SSH pública e privada e, em seguida, associe a chave pública ao seu usuário do IAM. Para obter mais informações sobre a configuração de SSH, consulte [Para conexões SSH no Linux, macOS ou Unix](#) e [Para conexões SSH no Windows](#).

Erro de acesso: mensagem “Taxa excedida” ou “429” ao se conectar a um repositório CodeCommit

Problema: quando você tenta se comunicar com um repositório do CodeCommit, é exibida uma mensagem informando "Taxa excedida" ou com um código de erro "429". A comunicação é reduzida significativamente ou há falha na comunicação.

Causa: Todas as chamadas para CodeCommit, seja de um aplicativo, do AWS CLI, de um cliente Git ou do Console de gerenciamento da AWS, estão sujeitas a um número máximo de solicitações por segundo e a solicitações ativas gerais. Você não pode exceder a taxa máxima de solicitação permitida para uma conta da Amazon Web Services em nenhuma Região da AWS. Se as solicitações excederem a taxa máxima, você receberá uma mensagem de erro e as chamadas seguintes terão um controle de utilização temporário para sua conta da Amazon Web Services. Durante o período de limitação, as conexões com o CodeCommit têm a velocidade reduzida e podem falhar.

Possíveis correções: tome medidas para reduzir o número de conexões, chamadas CodeCommit ou distribuir solicitações. Alguns abordagens a serem consideradas:

- Implemente a oscilação nas solicitações, especialmente em solicitações de sondagem periódicas

Se você tem um aplicativo que está fazendo pesquisas CodeCommit periodicamente e esse aplicativo está sendo executado em várias instâncias do Amazon EC2, introduza o jitter (uma quantidade aleatória de atraso) para que diferentes instâncias do Amazon EC2 não sejam pesquisadas no mesmo segundo. Recomendamos um número aleatório de 0 a 59 segundos para distribuir uniformemente os mecanismos de sondagem em um período de um minuto.

- Usar uma arquitetura baseada em eventos em vez de sondagem

Em vez da sondagem, use uma arquitetura baseada em eventos para que as chamadas só sejam feitas quando ocorrer um evento. Considere o uso CloudWatch de notificações de [AWS CodeCommit eventos](#) para acionar seu fluxo de trabalho.

- Implemente novas tentativas de erro e atrasos exponenciais e ações automatizadas do Git APIs

Novas tentativas em caso de erro e recuos exponenciais podem ajudar a limitar a taxa de chamadas. Cada SDK da AWS implementa a lógica de novas tentativas automáticas e algoritmos de recuo exponencial. Para Git push e Git pull automatizado, talvez seja necessário implementar sua própria lógica de novas tentativas. Para obter mais informações, consulte [Tentativas de erro e recuo exponencial](#) em. AWS

- Solicite um aumento da cota de CodeCommit serviço no AWS Support Center

Para receber um aumento de limite do serviço, você deve confirmar que já seguiu as sugestões oferecidas aqui, incluindo a implementação de novas tentativas em caso de erro ou métodos de recuo exponencial. Em sua solicitação, você também deve fornecer a Região da AWS conta da Amazon Web Services e o prazo afetado pelos problemas de limitação.

Solução de problemas de erros de configuração e AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar erros de configuração que você pode ver ao se conectar com AWS CodeCommit repositórios.

Tópicos

- [Erro de configuração: não é possível configurar AWS CLI credenciais no macOS](#)

Erro de configuração: não é possível configurar AWS CLI credenciais no macOS

Problema: Quando você `aws configure` executa para configurar o AWS CLI, você vê uma `ConfigParseError` mensagem.

Correções possíveis: a causa mais comum para esse erro é que o arquivo de credenciais já existe. Navegue até ~/.aws e procure um arquivo chamado `credentials`. Renomeie ou exclua esse arquivo e, em seguida, execute o comando `aws configure` novamente.

Solução de problemas de erros do console e AWS CodeCommit

As informações a seguir podem ajudá-lo a solucionar problemas de erros do console ao usar repositórios do AWS CodeCommit .

Tópicos

- [Erro de acesso: acesso à chave de criptografia negado para um CodeCommit repositório a partir do console ou AWS CLI](#)
- [Erro de criptografia: o repositório não pode ser descriptografado](#)
- [Erro do console: não é possível pesquisar o código em um CodeCommit repositório a partir do console](#)
- [Erro de exibição: não é possível visualizar um arquivo ou uma comparação entre arquivos](#)

Erro de acesso: acesso à chave de criptografia negado para um CodeCommit repositório a partir do console ou AWS CLI

Problema: Quando você tenta acessar CodeCommit pelo console ou pelo AWS CLI, aparece uma mensagem de erro contendo a frase `EncryptionKeyAccessDeniedException` ou `User is not authorized for the KMS default key for CodeCommit 'aws/codecommit' in your account`.

Possíveis correções: A causa mais comum desse erro é que sua conta da Amazon Web Services não está inscrita AWS Key Management Service, o que é necessário para CodeCommit. Abra o AWS KMS console, escolha chaves AWS gerenciadas e escolha **Começar agora**. Se você receber uma mensagem informando que não está inscrito no AWS Key Management Service serviço no momento, siga as instruções nessa página para se inscrever. Para obter mais informações sobre CodeCommit e AWS Key Management Service, consulte [AWS KMS e criptografia](#).

Erro de criptografia: o repositório não pode ser descriptografado

Problema: Quando você tenta acessar um CodeCommit repositório pelo console ou pelo AWS CLI, aparece uma mensagem de erro contendo a frase `Repository can't be decrypted`.

Possíveis correções: A causa mais comum desse erro é que a AWS KMS chave usada para criptografar e descriptografar dados desse repositório não está ativa ou está pendente de exclusão. É necessária uma chave ativa Chave gerenciada pela AWS ou gerenciada pelo cliente para CodeCommit. AWS Key Management Service Abra o AWS KMS console, escolha Chaves gerenciadas pela AWS ou Chaves gerenciadas pelo cliente e certifique-se de que a chave usada para o repositório esteja presente na Região da AWS local onde o repositório existe e que seu estado seja Ativo. Para obter mais informações sobre CodeCommit e AWS Key Management Service, consulte [AWS KMS e criptografia](#).

 Important

Se a chave usada para criptografar e descriptografar os dados do repositório tiver sido excluída permanentemente ou estiver inacessível, os dados nos repositórios criptografados com essa chave não poderão ser acessados.

Erro do console: não é possível pesquisar o código em um CodeCommit repositório a partir do console

Problema: quando você tenta navegar pelo conteúdo de um repositório do console, aparece uma mensagem de erro negando o acesso.

Possíveis correções: A causa mais comum desse erro é que uma política do IAM aplicada à sua conta da Amazon Web Services nega uma ou mais das permissões necessárias para navegar pelo código do CodeCommit console. Para obter mais informações sobre permissões de CodeCommit acesso e navegação, consulte [Autenticação e controle de acesso para AWS CodeCommit](#).

Erro de exibição: não é possível visualizar um arquivo ou uma comparação entre arquivos

Problema: Quando você tenta visualizar um arquivo ou uma comparação entre duas versões de um arquivo no CodeCommit console, aparece um erro informando que o arquivo ou a diferença é muito grande para ser exibido.

Possíveis correções: a causa mais comum desse erro é que o arquivo é muito grande para ser exibido, contém uma ou mais linhas que excedem o limite de caracteres de uma única linha em um arquivo ou que a diferença entre as duas versões do arquivo excede o limite de linhas. Para obter mais informações, consulte [Cotas](#). Para visualizar o arquivo ou as diferenças entre a versão

do arquivo, você pode abrir o arquivo localmente no seu IDE preferido, usar uma ferramenta de comparação do Git ou executar o comando `git diff`.

Solução de problemas, gatilhos e AWS CodeCommit

As seguintes informações podem ajudá-lo a solucionar problemas com gatilhos no AWS CodeCommit.

Tópicos

- [Erro de gatilho: um gatilho do repositório não é executado quando esperado](#)

Erro de gatilho: um gatilho do repositório não é executado quando esperado

Problema: um ou mais gatilhos configurados para um repositório parecem não ser executados ou não são executados como esperado.

Possíveis correções: se o alvo do gatilho for uma função do AWS Lambda, verifique se você configurou a política de recursos da função para ser acessada pelo CodeCommit.

Como alternativa, edite o gatilho e certifique-se de que os eventos para os quais deseja desencadear ações tenham sido selecionados e que as ramificações do gatilho incluam a ramificação na qual você deseja ver respostas às ações. Tente mudar as configurações do gatilho para All repository events e All branches, e depois teste o gatilho. Para obter mais informações, consulte [Editar de gatilhos para um repositório](#).

Ativação da depuração

Problema: eu quero ativar a depuração pra obter mais informações sobre meu repositório e como o Git está executando os comandos.

Correções possíveis: Tente o seguinte:

1. No terminal ou no prompt de comando, execute os seguintes comandos na máquina local antes de executar os comandos do Git:

No Linux, macOS ou Unix:

```
export GIT_TRACE_PACKET=1
```

```
export GIT_TRACE=1
export GIT_CURL_VERBOSE=1
```

No Windows:

```
set GIT_TRACE_PACKET=1
set GIT_TRACE=1
set GIT_CURL_VERBOSE=1
```

 Note

A configuração `GIT_CURL_VERBOSE` é útil somente para conexões HTTPS. O SSH não usa a biblioteca `libcurl`.

2. Para obter mais informações sobre seu repositório Git, recomendamos instalar a versão mais recente do [git-sizer](#). Siga as instruções para instalar o utilitário adequado ao seu sistema operacional e ambiente. Depois de instalado, na linha de comando ou no terminal, altere os diretórios para seu repositório local e execute o seguinte comando:

```
git-sizer --verbose
```

 Tip

Considere salvar a saída do comando em um arquivo para que você possa compartilhá-la facilmente com outras pessoas quando for solucionar problemas, principalmente ao longo do tempo.

AWS CodeCommit referência

Os tópicos de referência a seguir podem ajudar você a entender CodeCommit melhor o Git Regiões da AWS, os limites de serviço e muito mais.

Tópicos

- [Regiões e endpoints de conexão Git para AWS CodeCommit](#)
- [Usando AWS CodeCommit com interface VPC endpoints](#)
- [Cotas em AWS CodeCommit](#)
- [AWS CodeCommit referência de linha de comando](#)
- [Comandos básicos do Git](#)

Regiões e endpoints de conexão Git para AWS CodeCommit

Cada CodeCommit repositório está associado a um Região da AWS. CodeCommit oferece endpoints regionais para fazer suas solicitações ao serviço. Além disso, CodeCommit fornece pontos de extremidade de conexão Git para protocolos SSH e HTTPS em todas as regiões em que está disponível. CodeCommit

Todos os exemplos deste guia usam a mesma URL de endpoint para o Git no Leste dos EUA (Ohio): `git-codecommit.us-east-2.amazonaws.com`. No entanto, ao usar o Git e configurar suas conexões, certifique-se de escolher o endpoint de conexão do Git que corresponda ao que hospeda Região da AWS seu repositório. CodeCommit Por exemplo, se você desejar estabelecer uma conexão com um repositório no Leste dos EUA (Norte da Virgínia), use a URL de endpoint de `git-codecommit.us-east-1.amazonaws.com`. Isso também é válido para chamadas de API. Ao fazer conexões com um CodeCommit repositório com o AWS CLI ou o SDKs, certifique-se de usar o endpoint regional correto para o repositório.

Tópicos

- [Compatível Regiões da AWS com CodeCommit](#)
- [Endpoints de conexão do Git](#)
- [Impressões digitais do servidor para CodeCommit](#)

Compatível Regiões da AWS com CodeCommit

Você pode criar e usar CodeCommit repositórios da seguinte Regiões da AWS forma:

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- US West (Oregon)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Frankfurt)
- Europa (Estocolmo)
- Europa (Milão)
- África (Cidade do Cabo)
- Israel (Tel Aviv)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Jacarta)
- Oriente Médio (Emirados Árabes Unidos)
- Ásia-Pacífico (Seul)
- Asia Pacific (Osaka)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Hong Kong)
- América do Sul (São Paulo)
- Oriente Médio (Bahrein)
- Canadá (Central)
- China (Pequim)
- China (Ningxia)
- AWS GovCloud (Oeste dos EUA)

- AWS GovCloud (Leste dos EUA)

CodeCommit adicionou suporte ao padrão governamental da publicação 140-2 do Federal Information Processing Standard (FIPS) em algumas regiões. Para obter mais informações sobre a Norma federal de processamento de informações (Federal Information Processing Standard, FIPS) e endpoints da FIPS, consulte [Visão geral da publicação 140-2 da FIPS](#). Para os endpoints de conexão do Git que oferecem suporte à FIPS, consulte [Endpoints de conexão do Git](#).

Para obter mais informações sobre endpoints regionais AWS CLI, serviços e chamadas de API para CodeCommit, consulte [AWS CodeCommit Endpoints and Quotas](#).

Endpoints de conexão do Git

Use o seguinte URLs ao configurar conexões Git com repositórios: CodeCommit

Pontos de extremidade de conexão Git para AWS CodeCommit

Nome da região	Região	URL do endpoint	Protocolo
Leste dos EUA (Ohio)	us-east-2	<code>https://git-codecommit.us-east-2.amazonaws.com</code>	HTTPS
Leste dos EUA (Ohio)	us-east-2	<code>ssh://git-codecommit.us-east-2.amazonaws.com</code>	SSH
Leste dos EUA (Ohio)	us-east-2	<code>https://git-codecommit-fips.us-east-2.amazonaws.com</code>	HTTPS
Leste dos EUA (Norte da Virgínia)	us-east-1	<code>https://git-codecommit.us-east-1.amazonaws.com</code>	HTTPS
Leste dos EUA (Norte da Virgínia)	us-east-1	<code>ssh://git-codecommit.us-east-1.amazonaws.com</code>	SSH

Nome da região	Região	URL do endpoint	Protocolo
Leste dos EUA (Norte da Virgínia)	us-east-1	https://git-codecommit-fips.us-east-1.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	us-west-2	https://git-codecommit.us-west-2.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	us-west-2	ssh://git-codecommit.us-west-2.amazonaws.com	SSH
Oeste dos EUA (Oregon)	us-west-2	https://git-codecommit-fips.us-west-2.amazonaws.com	HTTPS
Oeste dos EUA (Norte da Califórnia)	us-west-1	https://git-codecommit.us-west-1.amazonaws.com	HTTPS
Oeste dos EUA (Norte da Califórnia)	us-west-1	ssh://git-codecommit.us-west-1.amazonaws.com	SSH
Oeste dos EUA (N. da Califórnia)	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

Nome da região	Região	URL do endpoint	Protocolo
Ásia-Pacífico (Tóquio)	ap-northeast-1	https://git-codecommit.ap-northeast-1.amazonaws.com	HTTPS
Ásia-Pacífico (Tóquio)	ap-northeast-1	ssh://git-codecommit.ap-northeast-1.amazonaws.com	SSH
Ásia-Pacífico (Singapura)	ap-southeast-1	https://git-codecommit.ap-southeast-1.amazonaws.com	HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	ssh://git-codecommit.ap-southeast-1.amazonaws.com	SSH
Ásia-Pacífico (Sydney)	ap-southeast-2	https://git-codecommit.ap-southeast-2.amazonaws.com	HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	ssh://git-codecommit.ap-southeast-2.amazonaws.com	SSH
Ásia-Pacífico (Jacarta)	ap-southeast-3	https://git-codecommit.ap-southeast-3.amazonaws.com	HTTPS
Ásia-Pacífico (Jacarta)	ap-southeast-3	ssh://git-codecommit.ap-southeast-3.amazonaws.com	SSH
Oriente Médio (Emirados Árabes Unidos)	me-central-1	https://git-codecommit.me-central-1.amazonaws.com	HTTPS

Nome da região	Região	URL do endpoint	Protocolo
Oriente Médio (Emirados Árabes Unidos)	me-central-1	ssh://git-codecommit.me-central-1.amazonaws.com	SSH
Europa (Frankfurt)	eu-central-1	https://git-codecommit.eu-central-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	ssh://git-codecommit.eu-central-1.amazonaws.com	SSH
Ásia-Pacífico (Seul)	ap-northeast-2	https://git-codecommit.ap-northeast-2.amazonaws.com	HTTPS
Ásia-Pacífico (Seul)	ap-northeast-2	ssh://git-codecommit.ap-northeast-2.amazonaws.com	SSH
América do Sul (São Paulo)	sa-east-1	https://git-codecommit.sa-east-1.amazonaws.com	HTTPS
América do Sul (São Paulo)	sa-east-1	ssh://git-codecommit.sa-east-1.amazonaws.com	SSH
Europa (Londres)	eu-west-2	https://git-codecommit.eu-west-2.amazonaws.com	HTTPS
Europa (Londres)	eu-west-2	ssh://git-codecommit.eu-west-2.amazonaws.com	SSH

Nome da região	Região	URL do endpoint	Protocolo
Ásia-Pacífico (Mumbai)	ap-south-1	https://git-codecommit.ap-south-1.amazonaws.com	HTTPS
Ásia-Pacífico (Mumbai)	ap-south-1	ssh://git-codecommit.ap-south-1.amazonaws.com	SSH
Ásia-Pacífico (Hyderabad)	ap-south-2	https://git-codecommit.ap-south-2.amazonaws.com	HTTPS
Ásia-Pacífico (Hyderabad)	ap-south-2	ssh://git-codecommit.ap-south-2.amazonaws.com	SSH
Canadá (Central)	ca-central-1	https://git-codecommit.ca-central-1.amazonaws.com	HTTPS
Canadá (Central)	ca-central-1	ssh://git-codecommit.ca-central-1.amazonaws.com	SSH
Canadá (Central)	ca-central-1	https://git-codecommit-fips.ca-central-1.amazonaws.com	HTTPS
Europa (Paris)	eu-west-3	https://git-codecommit.eu-west-3.amazonaws.com	HTTPS
Europa (Paris)	eu-west-3	ssh://git-codecommit.eu-west-3.amazonaws.com	SSH

Nome da região	Região	URL do endpoint	Protocolo
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	https://git-codecommit. us-gov-west-1.amaz onaws.com	HTTPS
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	ssh: //git-codecommit. us-gov-west-1.amaz onaws.com	SSH
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	https://git-codeco mdicas de luvas. us- gov-west-1.amaz onaws.com	HTTPS
AWS GovCloud (Leste dos EUA)	us-gov-east-1	https://git-codecommit. us-gov-east-1.amaz onaws.com	HTTPS
AWS GovCloud (Leste dos EUA)	us-gov-east-1	ssh: //git-codecommit. us-gov-east-1.amaz onaws.com	SSH
AWS GovCloud (Leste dos EUA)	us-gov-east-1	https://git-codeco mdicas de luvas. us- gov-east-1.amaz onaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	https://git-codeco mmit.eu-north-1.am azonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	ssh://git-codecomm it.eu-north-1.amaz onaws.com	SSH
Oriente Médio (Barém)	me-south-1	https://git-codeco mmit.me-south-1.am azonaws.com	HTTPS

Nome da região	Região	URL do endpoint	Protocolo
Oriente Médio (Barém)	me-south-1	ssh://git-codecommit.me-south-1.amazonaws.com	SSH
Ásia-Pacífico (Hong Kong)	ap-east-1	https://git-codecommit.ap-east-1.amazonaws.com	HTTPS
Ásia-Pacífico (Hong Kong)	ap-east-1	ssh://git-codecommit.ap-east-1.amazonaws.com	SSH
China (Pequim)	cn-north-1	https://git-codecommit.cn-north-1.amazonaws.com.cn	HTTPS
China (Pequim)	cn-north-1	ssh://git-codecommit.cn-north-1.amazonaws.com.cn	SSH
China (Ningxia)	cn-northwest-1	https://git-codecommit.cn-northwest-1.amazonaws.com.cn	HTTPS
China (Ningxia)	cn-northwest-1	ssh://git-codecommit.cn-northwest-1.amazonaws.com.cn	SSH
Europa (Milão)	eu-south-1	https://git-codecommit.eu-south-1.amazonaws.com	HTTPS
Europa (Milão)	eu-south-1	ssh://git-codecommit.eu-south-1.amazonaws.com	SSH

Nome da região	Região	URL do endpoint	Protocolo
Ásia-Pacífico (Osaka)	ap-northeast-3	https://git-codecommit.ap-northeast-3.amazonaws.com	HTTPS
Ásia-Pacífico (Osaka)	ap-northeast-3	ssh://git-codecommit.ap-northeast-3.amazonaws.com	SSH
África (Cidade do Cabo)	af-south-1	https://git-codecommit.af-south-1.amazonaws.com	HTTPS
África (Cidade do Cabo)	af-south-1	ssh://git-codecommit.af-south-1.amazonaws.com	SSH
Israel (Tel Aviv)	il-central-1	https://git-codecommit.il-central-1.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	ssh://git-codecommit.il-central-1.amazonaws.com	SSH

Impressões digitais do servidor para CodeCommit

A tabela a seguir lista as impressões digitais públicas para os endpoints de conexão do Git no CodeCommit. Essas impressões digitais do servidor são exibidas como parte do processo de verificação para adicionar um endpoint ao arquivo de hosts conhecido.

Impressões digitais públicas para CodeCommit

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.us-east-2.amazonaws.com	MD5	a9:6d:03:ed:08:42: 21:be:06:e1:e0:2a: d1:75:31:5e
git-codecommit.us-east-2.amazonaws.com	SHA256	3lB1W2g5xn/NA2Ck6d yeJIrQ0Wvn7n8UEs56 fG6ZlZQ
git-codecommit.us-east-1.amazonaws.com	MD5	a6:9c:7d:bc:35:f5: d4:5f:8b:ba:6f:c8: bc:d4:83:84
git-codecommit.us-east-1.amazonaws.com	SHA256	eLMY1j0DKA4uvDZc1/ KgtIayZANwX6t8+8is PtotBoY
git-codecommit.us-west-2.amazonaws.com	MD5	a8:68:53:e3:99:ac: 6e:d7:04:7e:f7:92: 95:77:a9:77
git-codecommit.us-west-2.amazonaws.com	SHA256	0pJx9SQpkbPUAHwy58 UVIq0IHcyo1fwCp00u VgcAWPo
git-codecommit.eu-west-1.amazonaws.com	MD5	93:42:36:ea:22:1f: f1:0f:20:02:4a:79: ff:ea:12:1d
git-codecommit.eu-west-1.amazonaws.com	SHA256	tKjRk0L8dmJyTmSbeS dN1S8F/f0iq13Rlvqg TOP1UyQ
git-codecommit.ap-northeast-1.amazonaws.com	MD5	8e:a3:f0:80:98:48: 1c:5c:6f:59:db:a7: 8f:6e:c6:cb

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.ap-northeast-1.amazonaws.com	SHA256	Xk/WeYD/K/bnBybzhi uu4dWpBJtXPf7E30jH U7se40w
git-codecommit.ap-southeast-1.amazonaws.com	MD5	65:e5:27:c3:09:68: 0d:8e:b7:6d:94:25: 80:3e:93:cf
git-codecommit.ap-southeast-1.amazonaws.com	SHA256	ZIsVa70VzxrtIf+Rk4 UbhPv6Es22mSB3uTBo jfPXIno
git-codecommit.ap-southeast-2.amazonaws.com	MD5	7b:d2:c1:24:e6:91: a5:7b:fa:c1:0c:35: 95:87:da:a0
git-codecommit.ap-southeast-2.amazonaws.com	SHA256	nYp+gHas80HY3DqbP4 yanCDFhqDVjseeFvBH EXqH2Ec
git-codecommit.ap-southeast-3.amazonaws.com	MD5	64:d9:e0:53:19:4f: a8:91:9a:c3:53:22: a6:a8:ed:a6
git-codecommit.ap-southeast-3.amazonaws.com	SHA256	ATdkGSFhpqIu7RqUVT /1RZo6MLxxxUW9NoDV MbAc/6g
git-codecommit.me-central-1.amazonaws.com	MD5	bd:fa:e2:f9:05:84: d6:39:6f:bc:d6:8d: fe:de:61:76
git-codecommit.me-central-1.amazonaws.com	SHA256	grceUDWubo4MzG1Noa KZKUfrgPvfN3ijli0n Qr11TZA

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.eu-central-1.amazonaws.com	MD5	74:5a:e8:02:fc:b2: 9c:06:10:b4:78:84: 65:94:22:2d
git-codecommit.eu-central-1.amazonaws.com	SHA256	MwGrkiEki8QkkBt1Ag XbYt0hoZYBnZF62VY5 RzGJEUy
git-codecommit.ap-northeast-2.amazonaws.com	MD5	9f:68:48:9b:5f:fc: 96:69:39:45:58:87: 95:b3:69:ed
git-codecommit.ap-northeast-2.amazonaws.com	SHA256	eegAPQrWY9YsYo9ZHI K0mxfXtBHzAZd8Eya 53Qcwko
git-codecommit.sa-east-1.amazonaws.com	MD5	74:99:9d:ff:2b:ef: 63:c6:4b:b4:6a:7f: 62:c5:4b:51
git-codecommit.sa-east-1.amazonaws.com	SHA256	kW+VKB0jpRaG/ZbXkg btMQbKgEDK7JnISV3S VoyCmzU
git-codecommit.us-west-1.amazonaws.com	MD5	3b:76:18:83:13:2c: f8:eb:e9:a3:d0:51: 10:32:e7:d1
git-codecommit.us-west-1.amazonaws.com	SHA256	gzauWTWDXDK2u5KuMMi 5vbKTmfyerdIwgSbzY B0DLpzg
git-codecommit.eu-west-2.amazonaws.com	MD5	a5:65:a6:b1:84:02: b1:95:43:f9:0e:de: dd:ed:61:d3

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.eu-west-2.amazonaws.com	SHA256	r0Rwz5k/IHp/QyrRnf iM9j02D5UEqMbtFNTu DG2hNbs
git-codecommit.ap-south-1.amazonaws.com	MD5	da:41:1e:07:3b:9e: 76:a0:c5:1e:64:88: 03:69:86:21
git-codecommit.ap-south-1.amazonaws.com	SHA256	hUKwnTj7+Xpx4Kddb6 p45j4RazIJ4IhAMD8k 29it0fE
git-codecommit.ap-south-2.amazonaws.com	MD5	bc:cc:9f:15:f8:f3: 58:a2:68:65:21:e2: 23:71:8d:ce
git-codecommit.ap-south-2.amazonaws.com	SHA256	Xe0CyZE0vgR5Xa2YUG qf+jn8/Ut7l7nX/Cms lSFNEig
git-codecommit.ca-central-1.amazonaws.com	MD5	9f:7c:a2:2f:8c:b5: 74:fd:ab:b7:e1:fd: af:46:ed:23
git-codecommit.ca-central-1.amazonaws.com	SHA256	Qz5puafQdANVprLlj6 r0Qyh4lCNsF6ob61dG cPtFS7w
git-codecommit.eu-west-3.amazonaws.com	MD5	1b:7f:97:dd:d7:76: 8a:32:2c:bd:2c:7b: 33:74:6a:76
git-codecommit.eu-west-3.amazonaws.com	SHA256	uw7c2FL564jVoFgtc+ ikzILnKBsZz7t9+CFd SJjKbLI

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.us-gov-west-1.amazonaws.com	MD5	9f:6c:19:3b:88:cd: e8:88:1b:9c:98:6a: 95:31:8a:69
git-codecommit.us-gov-west-1.amazonaws.com	SHA256	djXQoSIFcg8vHe0KVH 1xW/g0F9X37tWTqu4H kng75x4
git-codecommit.us-gov-east-1.amazonaws.com	MD5	00:8d:b5:55:6f:05: 78:05:ed:ea:cb:3f: e6:f0:62:f2
git-codecommit.us-gov-east-1.amazonaws.com	SHA256	fVb+R0z7qW7minenW+ rUpAABRCRBTCzmETAJ EQrg98
git-codecommit.eu-north-1.amazonaws.com	MD5	8e:53:d8:59:35:88: 82:fd:73:4b:60:8a: 50:70:38:f4
git-codecommit.eu-north-1.amazonaws.com	SHA256	b6KSK7xKq+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

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.ap-east-1.amazonaws.com	SHA256	LafadYwUYW8h0NoTRp objbNs9IRnbEwHtezD 3aAIBX0
git-codecommit.cn-north-1.amazonaws.com.cn	MD5	11:7e:2d:74:9e:3b: 94:a2:69:14:75:6f: 5e:22:3b:b3
git-codecommit.cn-north-1.amazonaws.com.cn	SHA256	IYUXxH20pTDsyYMLIp +JY8CTLS4UX+ZC5JVZ XPRaxc8
git-codecommit.cn-northwest-1.amazonaws.com.cn	MD5	2e:a7:fb:4c:33:ac: 6c:f9:aa:f2:bc:fb: 0a:7b:1e:b6
git-codecommit.cn-northwest-1.amazonaws.com.cn	SHA256	wqjd6eHd0+m0Bx+dCN uL0omUoCNjaDtZiEpW j5TmCfQ
git-codecommit.eu-south-1.amazonaws.com	MD5	b9:f6:5d:e2:48:92: 3f:a9:37:1e:c4:d0: 32:0e:fb:11
git-codecommit.eu-south-1.amazonaws.com	SHA256	1yXrWbCg3uQmJr11Xx B/ASR7ugW1Ysf5yzY0 JbudHsI
git-codecommit.ap-northeast-3.amazonaws.com	MD5	25:17:40:da:b9:d4: 18:c3:b6:b3:fb:ed: 1c:20:fe:29
git-codecommit.ap-northeast-3.amazonaws.com	SHA256	2B815B9F0AvwLnRxSV xUz4kDYmtEQUGGdQYP 8OQLXhA

Servidor	Tipo de hash de criptografia	Impressão digital
git-codecommit.af-south-1.amazonaws.com	MD5	21:a0:ba:d7:c1:d1: b5:39:98:8d:4d:7c: 96:f5:ca:29
git-codecommit.af-south-1.amazonaws.com	SHA256	C34ji3x/cnsDZjUpyN GXdE5pjHYimqJrQZ3l eTgqJHM
git-codecommit.il-central-1.amazonaws.com	MD5	04:74:89:16:98:7a: 61:b1:69:46:42:3c: d1:b4:ac:a9
git-codecommit.il-central-1.amazonaws.com	SHA256	uFxhp51kUWhleTLeYb xQVYm4RnNLNZ5Dbdm1 cgdS1/8

Usando AWS CodeCommit com interface VPC endpoints

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e CodeCommit. Você pode usar essa conexão para habilitar o CodeCommit a se comunicar com os seus recursos na VPC sem passar pela Internet pública.

O Amazon VPC é um AWS serviço que você pode usar para lançar AWS recursos em uma rede virtual que você define. Com a VPC, você tem controle sobre as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Com os VPC endpoints, o roteamento entre a VPC e os AWS serviços é gerenciado pela AWS rede, e você pode usar políticas do IAM para controlar o acesso aos recursos do serviço.

Para conectar sua VPC a CodeCommit, você define uma interface para a qual VPC endpoint. Um endpoint de interface é uma interface de rede elástica com um endereço IP privado que serve como ponto de entrada para o tráfego destinado a um serviço compatível AWS. O endpoint fornece conectividade confiável e escalável CodeCommit sem a necessidade de um gateway de internet, instância de tradução de endereços de rede (NAT) ou conexão VPN. Para obter mais informações, consulte [O que é a Amazon VPC?](#) no Guia do usuário da Amazon VPC.

 Note

Outros AWS serviços que fornecem suporte e se integram à VPC CodeCommit, como AWS CodePipeline, podem não oferecer suporte ao uso de endpoints da Amazon VPC para essa integração. Por exemplo, o tráfego entre CodePipeline e CodeCommit não pode ser restrito ao intervalo de sub-redes da VPC. Serviços que oferecem suporte à integração, como [AWS Cloud9](#), podem exigir serviços adicionais, como o AWS Systems Manager.

Os endpoints VPC da Interface são alimentados por AWS PrivateLink uma AWS tecnologia que permite a comunicação privada entre AWS serviços usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte [AWS PrivateLink](#).

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte [Conceitos básicos](#) no Guia do usuário da Amazon VPC.

Disponibilidade

CodeCommit atualmente oferece suporte a endpoints de VPC no seguinte: Regiões da AWS

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- US West (Oregon)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Frankfurt)
- Europa (Estocolmo)
- Europa (Milão)
- África (Cidade do Cabo)
- Israel (Tel Aviv)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Singapura)

- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Jacarta)
- Oriente Médio (Emirados Árabes Unidos)
- Ásia-Pacífico (Seul)
- Asia Pacific (Osaka)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Hong Kong)
- América do Sul (São Paulo)
- Oriente Médio (Bahrein)
- Canadá (Central)
- China (Pequim)
- China (Ningxia)
- AWS GovCloud (Oeste dos EUA)
- AWS GovCloud (Leste dos EUA)

Crie endpoints VPC para CodeCommit

Para começar a usar CodeCommit com sua VPC, crie uma interface VPC endpoint para. CodeCommit requer endpoints separados para operações do Git e CodeCommit para operações de API. Dependendo das necessidades da sua empresa, pode ser necessário criar mais de um VPC endpoint. Ao criar um VPC endpoint para CodeCommit, escolha AWS Serviços e, em Nome do serviço, escolha entre as seguintes opções:

- `com.amazonaws.region.git-codecommit`: escolha essa opção se quiser criar um VPC endpoint para operações do Git com repositórios. CodeCommit Por exemplo, escolha essa opção se seus usuários usam um cliente Git e comandos como `git pull`, `git commit`, e `git push` quando interagem com CodeCommit repositórios.
- `com.amazonaws.region.git-codecommit-fips`: escolha essa opção se quiser criar um VPC endpoint para operações do Git com CodeCommit repositórios que estejam em conformidade com a publicação 140-2 do Federal Information Processing Standard (FIPS), padrão do governo dos EUA.

Note

Os endpoints FIPS para Git não estão disponíveis em todas as regiões. AWS Para obter mais informações, consulte [Endpoints de conexão do Git](#).

- `com.amazonaws.region.codecommit`: escolha essa opção se quiser criar um VPC endpoint para operações de API. CodeCommit Por exemplo, escolha essa opção se seus usuários usarem a AWS CLI, a CodeCommit API ou AWS SDKs a CodeCommit para interagir em operações como `CreateRepositoryListRepositories`, `PutFile` e.
- `com.amazonaws.region.codecommit-fips`: escolha essa opção se quiser criar um VPC endpoint para operações de CodeCommit API que esteja em conformidade com a publicação 140-2 do Federal Information Processing Standard (FIPS), padrão do governo dos EUA.

Note

Os endpoints FIPS não estão disponíveis em todas as AWS regiões. Para obter mais informações, consulte a entrada AWS CodeCommit na Visão geral do [Federal Information Processing Standard \(FIPS\) 140-2](#).

Crie uma política de VPC endpoint para CodeCommit

Você pode criar uma política para endpoints da Amazon VPC, CodeCommit na qual você pode especificar:

- A entidade principal que pode realizar ações.
- As ações que podem ser executadas.
- Os recursos que podem ter ações executadas neles.

Por exemplo, uma empresa pode querer restringir o acesso a repositórios para o intervalo de endereços de rede em uma VPC. Você pode visualizar um exemplo desse tipo de política aqui: [Exemplo 3: permitir que um usuário conectado a partir de um intervalo de endereços IP especificado acesse um repositório](#). A empresa configurou dois endpoints da VPC do Git para a região Leste dos EUA (Ohio): `com.amazonaws.us-east-2.codecommit` e `com.amazonaws.us-east-2.git-codecommit-fips`. Eles querem permitir o envio de código para um CodeCommit repositório nomeado *MyDemoRepo* somente no endpoint compatível com FIPS. Para que isso seja aplicado,

eles configurariam uma política semelhante à seguinte no endpoint com `.amazonaws.us-east-2.codecommit` que especificamente nega ações de envio do Git:

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "codecommit:GitPush",
      "Effect": "Deny",
      "Resource": "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
      "Principal": "*"
    }
  ]
}
```

Important

A chave de condição global não `aws:VpcSourceIp` é compatível com CodeCommit repositórios nas políticas do IAM para `git push` comandos.

Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Cotas em AWS CodeCommit

A tabela a seguir descreve as cotas em CodeCommit. Para obter informações sobre cotas que podem ser alteradas, consulte [Endpoints e cotas do AWS CodeCommit](#). Para obter mais informações sobre como solicitar um aumento nas cotas de serviço, consulte [AWS Service Quotas](#). Para obter informações sobre as versões necessárias do Git e outros softwares, consulte [Compatibilidade para CodeCommit, Git e outros componentes](#).

Nomes de regra de aprovação e modelo de regra de aprovação

Qualquer combinação de letras, números, pontos, espaços, sublinhados e traços entre 1

	e 100 caracteres de comprimento. Os nomes diferenciam letras maiúsculas de minúsculas. Nomes não podem terminar em .git nem conter nenhum dos seguintes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Tamanho do conteúdo da regra de aprovação	3.000 caracteres
Tamanho da descrição do modelo de regra de aprovação	1.000 caracteres
Referências de destino do modelo de regra de aprovação	100
Modelos de regras de aprovação	1000 em um Região da AWS
Regras de aprovação em uma solicitação pull	Até 30 no máximo. Até 25 delas podem ser de modelos de regras de aprovação.
Regras de aprovação em uma solicitação pull criadas em um modelo de regra de aprovação	25
Aprovações em uma solicitação pull	200
Aprovadores em um grupo de aprovações	50

Nomes de ramificação	Qualquer combinação de caracteres permitido s entre 1 e 256 caracteres de comprimento, exceto que nomes de ramificações com exatamente 40 caracteres hexadecimais não são permitidos. Nomes de ramificação não podem: • começar nem terminar com uma barra (/) ou um ponto (.) • consistir em único caractere @ • conter dois ou mais pontos consecutivos (. .), barras (//) ou a seguinte combinação de caracteres: @{ • conter espaços ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ : Os nomes de ramificação são referências. Muitas limitações nos nomes de ramificação são baseadas no padrão de referência do Git. Para obter mais informações, consulte Git Internals e git-check-ref-format
Tamanho do comentário	Máximo de 10.240 caracteres.
Tamanho do arquivo compactado para inicializar um CodeCommit repositório no AWS CloudFormation	Máximo de 20 GB quando compactado.
Dados personalizados para triggers	Esse campo de string é limitado a 1.000 caracteres. Ele não pode ser usado para passar nenhum parâmetro dinâmico.

Exibir no console	Um arquivo ou uma comparação entre arquivos pode não estar visível no console se: • O arquivo tem mais de 2 MB • O arquivo contém mais de 25.000 caracteres em uma única linha • Uma comparação contém mais de 6.500 linhas totais de diferenças
Os endereços de e-mail em confirmações feitas no console	Qualquer combinação dos caracteres permitidos entre 1 e 256 caracteres. Os endereços de e-mail não foram validados.

Caminhos do arquivo	Qualquer combinação dos caracteres permitidos entre 1 e 4,096 caracteres. Os caminhos de arquivo devem ser um nome exclusivo que especifica o arquivo e o local exato dele. Os caminhos de arquivo não podem exceder 20 diretórios de extensão. Além disso, os caminhos de arquivo não podem: • conter sequências de caracteres vazias • ser um caminho de arquivo relativo • incluir qualquer uma das seguintes combinações: /./ /../ // • terminar com barra no final ou barra invertida Os nomes de arquivo e caminhos devem ser totalmente qualificados. O nome e o caminho de um arquivo no seu computador local devem seguir os padrões do sistema operacional em questão. Ao especificar o caminho para um arquivo em um CodeCommit repositório, use os padrões do Amazon Linux.
Tamanho do arquivo	Máximo de 6 MB para qualquer arquivo individual ao usar o CodeCommit console ou AWS CLI ou APIs

Tamanho do blob do Git	Máximo de 2 GB.  Note Não há limite no número ou tamanho total de todos os arquivos em uma única confirmação, desde que os metadados não ultrapassem 6 MB e um único blob não exceda 2 GB.
Exibição gráfica de ramificações no Commit Visualizer	35 por página. Se houver mais de 35 ramificações em uma única página, o gráfico não será exibido.
Metadados de uma confirmação	Máximo de 20 MB para os metadados combinados de uma confirmação (por exemplo, a combinação de informações do autor, data, lista de confirmações principal e mensagens de confirmação) ao usar o CodeCommit console ou o. APIs AWS CLI  Note Não há limite no número ou tamanho total de todos os arquivos em uma única confirmação, desde que os metadados não ultrapassem 6 MB, um arquivo individual não ultrapasse 6 MB e um único blob não ultrapasse 2 GB.
Número de arquivos em uma confirmação	Máximo de 100.
Número de solicitações pull abertas	Máximo de 1.000.
Número de referências em um único push	Máximo de 4.000, incluindo criação, exclusão e atualização. Não há limites para o número total de referências no repositório.

Número de repositórios	Máximo de 5.000 por conta da Amazon Web Services. Esse limite pode ser alterado. Para obter mais informações, consulte Endpoints e cotas do AWS CodeCommit e AWS Service Quotas .
Número de triggers em um repositório	Máximo de 10.

Regiões

CodeCommit está disponível no seguinte Regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- US West (Oregon)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Frankfurt)
- Europa (Estocolmo)
- Europa (Milão)
- África (Cidade do Cabo)
- Israel (Tel Aviv)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Jacarta)
- Oriente Médio (Emirados Árabes Unidos)
- Ásia-Pacífico (Seul)
- Asia Pacific (Osaka)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Hong Kong)
- América do Sul (São Paulo)
- Oriente Médio (Bahrein)
- Canadá (Central)
- China (Pequim)
- China (Ningxia)
- AWS GovCloud (Oeste dos EUA)

	• AWS GovCloud (Leste dos EUA) Para obter mais informações, consulte Regiões e endpoints de conexão do Git.
Descrições de repositório	Qualquer combinação de caracteres entre 0 e 1.000 caracteres. As descrições de repositório são opcionais.
Nomes de repositório	Qualquer combinação de letras, números, pontos, sublinhados e traços entre 1 e 100 caracteres. Os nomes diferenciam letras maiúsculas de minúsculas. Nomes de repositório não podem terminar em .git nem conter nenhum dos seguintes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Nomes de chave de tag de repositório	Qualquer combinação de letras, números, espaços e caracteres Unicode permitidos em UTF-8, entre 1 e 128 caracteres de comprimento. Os caracteres permitidos são + - = . _ : / @ Os nomes de chaves de tag devem ser exclusivos, e cada chave só pode ter um valor. Uma tag não pode: • começar com aws : • consistir apenas de espaços • terminar com um espaço • conter emojis ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;

Valores de tag de repositório	Qualquer combinação de letras, números, espaços e caracteres Unicode permitidos em UTF-8, entre 1 e 256 caracteres de comprimento. Os caracteres permitidos são + - = . _ : / @ Uma chave pode ter apenas um valor, mas várias chaves podem ter o mesmo valor. Uma tag não pode: • consistir apenas de espaços • terminar com um espaço • conter emojis ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Tags de repositório	As tags diferenciam letras maiúsculas de minúsculas. Máximo de 50 por recurso. Não são permitidos nomes de tags com exatamente 40 caracteres hexadecimais.
Nomes de trigger	Qualquer combinação de letras, números, pontos, sublinhados e traços entre 1 e 100 caracteres. Nomes do trigger não podem conter espaços ou vírgulas.
Os nomes de usuário em confirmações feitas no console	Qualquer combinação dos caracteres permitidos entre 1 e 1,024 caracteres.

AWS CodeCommit referência de linha de comando

Esta referência ajuda a aprender a usar o AWS CLI.

Para instalar e configurar o AWS CLI

1. Em sua máquina local, baixe e instale AWS CLI o. Esse é um pré-requisito para interagir com a linha CodeCommit de comando. Recomendamos instalar a AWS CLI versão 2. É a versão

principal mais recente do AWS CLI e oferece suporte a todos os recursos mais recentes. É a única versão do AWS CLI que suporta o uso de uma conta raiz, acesso federado ou credenciais temporárias com. `git-remote-codecommit`

Para obter mais informações, consulte [Como configurar a interface de linha de AWS comando](#).

 Note

CodeCommit funciona somente com AWS CLI as versões 1.7.38 e posteriores.

Como prática recomendada, instale ou atualize AWS CLI para a versão mais recente disponível. Para determinar qual versão do AWS CLI você instalou, execute o `aws --version` comando.

Para atualizar uma versão mais antiga do AWS CLI para a versão mais recente, consulte [Instalando AWS Command Line Interface](#) o.

2. Execute esse comando para verificar se os CodeCommit comandos do AWS CLI estão instalados.

```
aws codecommit help
```

Esse comando retorna uma lista de CodeCommit comandos.

3. Configure o AWS CLI com um perfil usando o `configure` comando, do seguinte modo:.

```
aws configure
```

Quando solicitado, especifique a chave de AWS acesso e a chave de acesso AWS secreta do usuário do IAM com CodeCommit a qual usar. Além disso, certifique-se de especificar Região da AWS onde o repositório existe, com `us-east-2`. Quando solicitado pelo formato de saída padrão, especifique `json`. Por exemplo, se você estiver configurando um perfil para um usuário do IAM:

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

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

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

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

Para obter mais informações sobre como criar e configurar perfis para usar com o AWS CLI, consulte o seguinte:

- [Perfis nomeados](#)
- [Usando uma função do IAM no AWS CLI](#)
- [Comando set](#)
- [Conectando-se a AWS CodeCommit repositórios com credenciais rotativas](#)

Para se conectar a um repositório ou recurso em outra Região da AWS, você deve reconfigurar o AWS CLI com o nome de região padrão. Os nomes de região padrão compatíveis CodeCommit incluem:

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

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

Para obter mais informações sobre CodeCommit e Região da AWS, consulte [Regiões e endpoints de conexão do Git](#). Para obter mais informações sobre o IAM, chaves de acesso e chaves secretas, consulte [Como consigo credenciais?](#) e [Gerenciar chaves de acesso para usuário do IAM](#). Para obter mais informações sobre os perfis AWS CLI e, consulte [Perfis nomeados](#).

Para ver uma lista de todos os CodeCommit comandos disponíveis, execute o seguinte comando:

```
aws codecommit help
```

Para ver informações sobre um CodeCommit comando, execute o comando a seguir, onde *command-name* está o nome do comando (por exemplo, create-repository):

```
aws codecommit command-name help
```

Veja a seguir descrições e exemplos de uso dos comandos no AWS CLI:

- [associate-approval-rule-template-com repositório](#)
- [batch-associate-approval-rule-template-with-repositories](#)
- [batch-disassociate-approval-rule-template-from-repositories](#)
- [batch-describe-merge-conflicts](#)
- [batch-get-commits](#)
- [batch-get-repositories](#)
- [create-approval-rule-template](#)
- [create-branch](#)

- [create-commit](#)
- [create-pull-request](#)
- [create-pull-request-approval-regra](#)
- [create-repository](#)
- [create-unreferenced-merge-commit](#)
- [delete-approval-rule-template](#)
- [delete-branch](#)
- [delete-comment-content](#)
- [delete-file](#)
- [delete-repository](#)
- [describe-merge-conflicts](#)
- [delete-pull-request-approval-regra](#)
- [describe-pull-request-events](#)
- [disassociate-pull-request-approval-rule-template-from-repository](#)
- [evaluate-pull-request-approval-regras](#)
- [get-approval-rule-template](#)
- [get-blob](#)
- [get-branch](#)
- [get-comment](#)
- [get-comment-reactions](#)
- [get-comments-for-compared-cometer](#)
- [get-comments-for-pull-solicitação](#)
- [get-commit](#)
- [get-differences](#)
- [get-merge-commit](#)
- [get-merge-conflicts](#)
- [get-merge-options](#)
- [get-pull-request](#)
- [get-pull-request-approval-estados](#)
- [get-pull-request-override-estado](#)

- [get-repository](#)
- [get-repository-triggers](#)
- [list-approval-rule-templates](#)
- [list-associated-approval-rule-templates-for-repository](#)
- [list-branches](#)
- [list-pull-requests](#)
- [list-repositories](#)
- [list-repositories-for-approval-modelo de regra](#)
- [list-tags-for-resource](#)
- [merge-branches-by-fast-para frente](#)
- [merge-branches-by-squash](#)
- [merge-branches-by-three-caminho](#)
- [merge-pull-request-by-avanço rápido](#)
- [merge-pull-request-by-abóbora](#)
- [merge-pull-request-by-três vias](#)
- [override-pull-request-approval-regras](#)
- [post-comment-for-compared-cometer](#)
- [post-comment-for-pull-solicitação](#)
- [post-comment-reply](#)
- [put-comment-reaction](#)
- [put-file](#)
- [put-repository-triggers](#)
- [tag-resource](#)
- [test-repository-triggers](#)
- [untag-resource](#)
- [update-approval-rule-template-conteúdo](#)
- [update-approval-rule-template-descrição](#)
- [update-approval-rule-template-nome](#)
- [update-comment](#)
- [update-default-branch](#)

- [update-pull-request-approval-conteúdo da regra](#)
- [update-pull-request-approval-estado](#)
- [update-pull-request-description](#)
- [update-pull-request-status](#)
- [update-pull-request-title](#)
- [update-repository-description](#)
- [update-repository-name](#)

Comandos básicos do Git

Você pode usar o Git para trabalhar com um repositório local e o CodeCommit repositório ao qual você conectou o repositório local.

Veja alguns exemplos básicos de comandos do Git muito usados abaixo.

Para obter mais opções, consulte a documentação do Git.

Tópicos

- [Variáveis de configuração](#)
- [Repositórios remotos](#)
- [Confirmações](#)
- [Ramificações](#)
- [Tags](#)

Variáveis de configuração

Lista todas as variáveis de configuração.	<code>git config --list</code>
Lista somente variáveis de configuração locais.	<code>git config --local -l</code>
Lista somente variáveis de configuração do sistema.	<code>git config --system -l</code>
Lista somente variáveis de configuração globais.	<code>git config --global -l</code>

Define uma variável de configuração no arquivo de configuração especificado.

```
git config [--local | --global | --system] variable-name variable-value
```

Define o nome da ramificação padrão como principal para todos os repositórios locais quando é feita uma confirmação inicial em um repositório que ainda não tem uma ramificação padrão

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

Edita um arquivo de configuração diretamente. Também pode ser usada para descobrir o local de um arquivo de configuração específico. Para sair do modo de edição, digite :q (para sair sem salvar alterações) ou :wq (para salvar alterações e sair) e pressione Enter.

```
git config [--local | --global | --system] --edit
```

Repositórios remotos

Inicializa um repositório local em preparação para conectá-lo a um CodeCommit repositório.

```
git init
```

Pode ser usado para configurar uma conexão entre um repositório local e um repositório remoto (como um repositório) usando o apelido especificado que o CodeCommit repositório local tem para o repositório e a URL especificada para o CodeCommit repositório.

```
git remote add remote-name remote-url
```

Cria um repositório local fazendo uma cópia de um CodeCommit repositório na URL especificada, na subpasta especificada da pasta atual na máquina local. Esse comando também cria uma ramificação de rastreamento remoto para cada ramificação no CodeCommit repositório

```
git clone remote-url local-subfolder-name
```

clonado e cria e verifica uma ramificação inicial que é bifurcada da ramificação padrão atual no repositório clonado. CodeCommit

Mostra o apelido que o repositório local usa para o CodeCommit repositório.

```
git remote
```

Mostra o apelido e a URL que o repositório local usa para buscas e envios para o repositório. CodeCommit

```
git remote -v
```

Envia os commits finalizados do repositório local para o repositório, usando o apelido especificado que o CodeCommit repositório local tem para o repositório e a ramificação especificada. CodeCommit Também configura informações de rastreamento upstream do repositório local durante o envio via push.

```
git push -u remote-name branch-name
```

Envia os commits finalizados do repositório local para o CodeCommit repositório depois que as informações de rastreamento upstream são definidas.

```
git push
```

Extrai os commits finalizados do repositório para o repositório local, usando o apelido especificado que o CodeCommit repositório local tem para o repositório e a ramificação especificada CodeCommit

```
git pull remote-name branch-name
```

Extrai os commits finalizados do repositório para o CodeCommit repositório local após a definição das informações de rastreamento upstream.

```
git pull
```


Desconecta o repositório local do CodeCommit repositório, usando o apelido especificado que o repositório local tem para o repositório. CodeCommit

```
git remote rm remote-name
```

Confirmações

Mostra o que foi ou não adicionado à confirmação pendente no repositório local.

```
git status
```

Mostra o que foi ou não adicionado à confirmação pendente no repositório local em um formato conciso.

```
git status -sb
```

(M = modificado, A = adicionado, D = excluído, e assim por diante)

Mostra alterações entre a confirmação pendente e a confirmação mais recente no repositório local.

```
git diff HEAD
```

Adiciona arquivos específicos à confirmação pendente no repositório local.

```
git add [file-name-1 file-name-2  
file-name-N | file-pattern ]
```

Adiciona todos os arquivos novos, modificados e excluídos à confirmação pendente no repositório local.

```
git add
```

Começa a finalizar a confirmação pendente no repositório local, que exibe um editor para fornecer uma mensagem de confirmação. Após a mensagem ser inserida, a confirmação pendente é finalizada.

```
git commit
```

Finaliza a confirmação pendente no repositório local, incluindo a especificação de uma mensagem de confirmação ao mesmo tempo.

```
git commit -m "Some meaningful  
commit comment"
```

Lista confirmações recentes no repositório local.	<code>git log</code>
Lista confirmações recentes no repositório local em formato de gráfico.	<code>git log --graph</code>
Lista confirmações recentes no repositório local em formato condensado predefinido.	<code>git log --pretty=oneline</code>
Lista confirmações recentes no repositório local em formato condensado predefinido, com um gráfico.	<code>git log --graph --pretty=oneline</code>
Lista confirmações recentes no repositório local em formato personalizado, com um gráfico. (Para obter mais opções, consulte Sobre o histórico: commits na documentação do Git.)	<code>git log --graph --pretty=format:"%H (%h) : %cn : %ar : %s"</code>

Ramificações

Lista todas as ramificações no repositório local com um asterisco (*) exibido ao lado da sua ramificação atual.	<code>git branch</code>
Extraí informações sobre todas as ramificações existentes no CodeCommit repositório para o repositório local.	<code>git fetch</code>
Lista todas as ramificações no repositório local e ramificações de rastreamento remoto no repositório local.	<code>git branch -a</code>
Lista somente ramificações de rastreamento remoto no repositório local.	<code>git branch -r</code>

Cria uma nova ramificação no repositório local usando o nome de ramificação especificado.	<code>git branch <i>new-branch-name</i></code>
Altera para outra ramificação no repositório local usando o nome de ramificação especificado.	<code>git checkout <i>other-branch-name</i></code>
Cria uma nova ramificação no repositório local usando o nome de ramificação especificado e alterna para ela.	<code>git checkout -b <i>new-branch-name</i></code>
Envia uma nova ramificação do repositório local para o CodeCommit repositório usando o apelido especificado que o repositório local tem para o CodeCommit repositório e o nome da ramificação especificada. Também configura informações de rastreamento upstream da ramificação no repositório local durante o envio via push.	<code>git push -u <i>remote-name new-branch-name</i></code>
Cria uma nova ramificação no repositório local usando o nome de ramificação especificado. Em seguida, conecta a nova ramificação no repositório local a uma ramificação existente no CodeCommit repositório, usando o apelido especificado que o repositório local tem para o CodeCommit repositório e o nome da ramificação especificada.	<code>git branch --track <i>new-branch-name remote-name /remote-branch-name</i></code>
Mescla alterações de outra ramificação no repositório local à ramificação atual no repositório local.	<code>git merge <i>from-other-branch-name</i></code>
Exclui uma ramificação no repositório local, a menos que ela contenha trabalho que ainda não foi mesclado.	<code>git branch -d <i>branch-name</i></code>

Exclui uma ramificação no CodeCommit repositório usando o apelido especificado que o repositório local tem para o CodeCommit repositório e o nome da ramificação especificada. (Observe o uso de dois pontos (:).)

```
git push remote-name :branch-name
```

Tags

Lista todas as tags no repositório local.

```
git tag
```

Extrai todas as tags do CodeCommit repositório para o repositório local.

```
git fetch --tags
```

Mostra informações sobre uma tag específica no repositório local.

```
git show tag-name
```

Cria uma tag "leve" no repositório local.

```
git tag tag-name commit-id-to-point-tag-at
```

Envia uma tag específica do repositório local para o CodeCommit repositório usando o apelido especificado que o repositório local tem para o CodeCommit repositório e o nome da tag especificada.

```
git push remote-name tag-name
```

Envia todas as tags do repositório local para o CodeCommit repositório usando o apelido especificado que o repositório local tem para o repositório. CodeCommit

```
git push remote-name --tags
```

Exclui uma tag no repositório local.

```
git tag -d tag-name
```

Exclui uma tag no CodeCommit repositório usando o apelido especificado que o repositório local tem para o CodeCommit repositório e o nome da tag especificada. (Observe o uso de dois pontos (:).)

```
git push remote-name :tag-name
```

AWS CodeCommit Histórico do documento do Guia do Usuário

A tabela a seguir descreve mudanças importantes na documentação do CodeCommit. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

- Versão da API: 13/04/2015

Alteração	Descrição	Data
AWS CodeCommit agora está disponível para novos clientes.	AWS CodeCommit agora está disponível para novos clientes. Os clientes atuais do AWS CodeCommit podem continuar usando o serviço normalmente.	25 de novembro de 2025
AWS CodeCommit não está mais disponível para novos clientes.	AWS CodeCommit não está mais disponível para novos clientes. Os clientes atuais do AWS CodeCommit podem continuar usando o serviço normalmente. Saiba mais	25 de julho de 2024
Informações de solução de problemas atualizadas para o assistente de credenciais	Foram adicionadas informações sobre a solução de problemas causados por arquivos.gitconfig incompletos ou mal configurados.	9 de julho de 2024
Informações de solução de problemas atualizadas para git-remote-codecommit	Foram adicionadas informações extras sobre a solução de problemas de erros em ambientes gerenciados externamente.	9 de julho de 2024

[Referências removidas a AWS Toolkit for Eclipse](#)

AWS Toolkit for Eclipse chegou ao fim da vida útil e não é mais suportado. As referências a esse kit de ferramentas tenham sido removidas.

9 de julho de 2024

[CodeCommit agora suporta o uso de chaves gerenciadas pelo cliente](#)

Agora você pode usar uma chave gerenciada pelo cliente ou uma Chave gerenciada pela AWS para criptografar e descriptografar dados em um repositório. Consulte mais informações em [AWS KMS e criptografia para repositórios do AWS CodeCommit](#), [Criar um repositório do AWS CodeCommit](#) e [Alteração das configurações de repositório do AWS CodeCommit](#).

21 de dezembro de 2023

[CodeCommit agora está disponível em Israel \(Tel Aviv\);](#)

CodeCommit agora está disponível em Israel (Tel Aviv). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

28 de agosto de 2023

[Alterações nas políticas gerenciadas para CodeCommit](#)

As AWSCode CommitFullAccess políticas AWSCode CommitPowerUser e foram atualizadas com uma permissão adicional. Para obter mais informações, consulte [CodeCommit atualizações nas políticas AWS gerenciadas](#).

16 de maio de 2023

[CodeCommit agora está disponível em três adicionais Regiões da AWS](#)

CodeCommit agora está disponível em mais três Regiões da AWS: Ásia-Pacífico (Jacarta), Oriente Médio (EAU) e Ásia-Pacífico (Hyderabad). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

28 de fevereiro de 2023

[CodeCommit agora está disponível na África \(Cidade do Cabo\)](#)

CodeCommit agora está disponível em um adicional Região da AWS: África (Cidade do Cabo). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

15 de setembro de 2021

[Alterações nas políticas gerenciadas para CodeCommit](#)

Detalhes sobre as atualizações das políticas AWS CodeCommit gerenciadas já estão disponíveis. Para obter mais informações, consulte [CodeCommit atualizações nas políticas AWS gerenciadas](#).

18 de agosto de 2021

[CodeCommit agora está disponível na Ásia-Pacífico \(Osaka\)](#)

CodeCommit agora está disponível em um adicional Região da AWS: Ásia-Pacífico (Osaka). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

14 de abril de 2021

[AWS CloudFormation e AWS Cloud Development Kit \(AWS CDK\) altera o comportamento de nomenclatura para ramificações padrão em CodeCommit](#)

Os repositórios criados usando CloudFormation ou AWS CDK com uma confirmação inicial de código agora usam o nome padrão da ramificação principal. Essa alteração não afeta repositórios ou ramificações existentes. Os clientes que usam clientes Git locais para criar suas confirmações iniciais têm um nome de ramificação padrão que segue a configuração desses clientes Git. Para obter mais informações, consulte [Criação de CodeCommit recursos com AWS CloudFormation](#).

4 de março de 2021

[CodeCommit altera o comportamento de nomenclatura para ramificações padrão](#)

Em 19 de janeiro de 2021, o nome padrão da ramificação criado por uma confirmação inicial em um CodeCommit repositório é main. Essa alteração não afeta repositórios ou ramificações existentes. Os clientes que usam clientes Git locais para criar suas confirmações iniciais têm um nome de ramificação padrão que segue a configuração desses clientes Git. Para obter mais informações, consulte [Trabalhar com ramificações](#), [Criar uma confirmação](#) e [Alterar as configurações da ramificação](#).

19 de janeiro de 2021

[CodeCommit agora está disponível na Europa \(Milão\)](#)

CodeCommit agora está disponível em um adicional Região da AWS: Europa (Milão). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

16 de setembro de 2020

[CodeCommit adiciona suporte para reações de emoji aos comentários](#)

CodeCommit agora suporta reagir aos comentários de outros usuários com emojis. Para obter mais informações, consulte [Comentar em uma confirmação](#) e [Revisar uma solicitação pull](#).

24 de junho de 2020

[CodeCommit agora disponível na China \(Pequim\) e na China \(Ningxia\)](#)

CodeCommit agora está disponível em mais dois Regiões da AWS: China (Pequim) e China (Ningxia). Para obter mais informações, consulte [Regiões e endpoints de conexão do Git](#).

23 de abril de 2020

[CodeCommit adiciona suporte para git-remote-codecommit](#)

CodeCommit suporta conexões com CodeCommit repositórios via HTTPS comgit-remote-codecommit, um utilitário que modifica o Git. Essa é a abordagem recomendada para conexões de acesso federado ou temporário aos CodeCommit repositórios. Você também pode usar o git-remote-codecommit com um usuário do IAM. git-remote-codecommit não requer a configuração de credenciais do Git para o usuário. Para obter mais informações, consulte [Etapas de configuração para conexões HTTPS AWS CodeCommit com git-remote-codecommit](#).

4 de março de 2020

[CodeCommit suporta tags de sessão](#)

CodeCommit suporta o uso de tags de sessão, que são atributos de pares de valores-chave que você transmite quando assume uma função do IAM, usa credenciais temporárias ou federa um usuário em (). AWS Security Token Service AWS STS. Você pode usar as informações fornecidas nessas tags para facilitar a identificação de quem fez uma alteração ou causou um evento. Para obter mais informações, consulte [Monitoramento do CodeCommit](#) e [Como usar tags para fornecer informações de identidade no CodeCommit](#).

19 de dezembro de 2019

[CodeCommit está disponível na Ásia-Pacífico \(Hong Kong\)](#)

Agora você pode usar CodeCommit na Ásia-Pacífico (Hong Kong). Para obter mais informações, incluindo sobre endpoints de conexão do Git, consulte [Regiões](#).

11 de dezembro de 2019

[CodeCommit suporta Amazon CodeGuru Reviewer](#)

CodeCommit oferece suporte ao Amazon CodeGuru Reviewer, um serviço automatizado de revisão de código que usa análise de programas e aprendizado de máquina para detectar problemas comuns e recomendar correções em seu código Java ou Python. Para obter mais informações, consulte [Associar ou desassociar um repositório com o Amazon CodeGuru Reviewer](#) e [Trabalhar com pull requests](#).

3 de dezembro de 2019

[CodeCommit suporta regras de aprovação](#)

Agora você pode usar regras de aprovação para ajudá-lo a personalizar seus fluxos de trabalho de desenvolvimento em vários repositórios, de modo que diferentes ramificações tenham os níveis apropriados de aprovações e controles para solicitações pull. Para obter mais informações, consulte [Trabalhar com modelos de regra de aprovação](#) e [Trabalhar com solicitações pull](#).

20 de novembro de 2019

[CodeCommit suporta regras de notificação](#)

Agora você pode usar regras de notificação para informar os usuários sobre alterações importantes nos repositórios. A funcionalidade desse recurso substitui as notificações criadas antes de 5 de novembro de 2019. Para obter mais informações, consulte [Criar uma regra de notificação](#).

5 de novembro de 2019

[CodeCommit está disponível no Oriente Médio \(Bahrein\)](#)

Agora você pode usar CodeCommit no Oriente Médio (Bahrein). Para obter mais informações, incluindo sobre endpoints de conexão do Git, consulte [Regiões](#).

30 de outubro de 2019

[CodeCommit adiciona suporte para recuperar informações sobre vários commits](#)

Você pode obter informações sobre vários commits usando o batch-get-commits comando no AWS CLI. Para obter mais informações, consulte [Visualizar detalhes de confirmação](#).

15 de agosto de 2019

[CodeCommit está disponível na Europa \(Estocolmo\)](#)

Agora você pode usar CodeCommit na Europa (Estocolmo). Para obter mais informações, incluindo sobre endpoints de conexão do Git, consulte [Regiões](#).

31 de julho de 2019

[CodeCommit adiciona suporte para marcar repositórios no console CodeCommit](#)

Agora você pode adicionar, gerenciar e remover tags de um repositório para ajudá-lo a gerenciar seus AWS recursos a partir do CodeCommit console. Para obter mais informações, consulte [Marcar um repositório](#).

2 de julho de 2019

[CodeCommit adiciona suporte para estratégias adicionais de mesclagem do Git](#)

Agora, você pode escolher entre as estratégias de mesclagem do Git ao mesclar solicitações pull no CodeCommit. Você também pode resolver conflitos de mesclagem no CodeCommit console. Para obter mais informações, consulte [Trabalhar com solicitações pull](#).

10 de junho de 2019

[CodeCommit está disponível em AWS GovCloud \(Leste dos EUA\)](#)

Agora você pode usar CodeCommit em AWS GovCloud (Leste dos EUA). Para obter mais informações, incluindo sobre endpoints de conexão do Git, consulte [Regiões](#).

31 de maio de 2019

[CodeCommit adiciona suporte para marcar repositórios](#)

Agora você pode adicionar, gerenciar e remover tags de um repositório para ajudá-lo a gerenciar seus AWS recursos. Para obter mais informações, consulte [Marcar um repositório](#).

30 de maio de 2019

Encontrar recursos no console	Agora você pode pesquisar rapidamente recursos, como repositórios, projetos de compilação, aplicativos de implantação e pipelines. Selecione Go to resource (Acessar recurso) ou pressione a tecla / e digite o nome do recurso. Para obter mais informações, consulte Tutorial do CodeCommit.	14 de maio de 2019
CodeCommit está disponível em AWS GovCloud (Oeste dos EUA)	Agora você pode usar CodeCommit em AWS GovCloud (Oeste dos EUA). Para obter mais informações, incluindo sobre endpoints de conexão do Git, consulte Regiões.	18 de abril de 2019
CodeCommit adiciona suporte para endpoints Amazon VPC	Agora você pode estabelecer uma conexão privada entre sua VPC e CodeCommit. Para obter mais informações, consulte Como usar CodeCommit com a interface VPC Endpoints.	7 de março de 2019
CodeCommit adiciona uma nova API	CodeCommit adicionou uma API para criar commits. Para obter mais informações, consulte Criar uma confirmação.	20 de fevereiro de 2019

Atualização de conteúdo	O conteúdo deste guia foi atualizado com pequenas correções e orientações de problemas adicionais.	2 de janeiro de 2019
Atualização de conteúdo	O conteúdo deste guia foi atualizado para oferecer suporte à nova experiência CodeCommit do console.	30 de outubro de 2018
CodeCommit e o padrão federal de processamento de informações (FIPS)	CodeCommit adicionou suporte ao padrão governamental da publicação 140-2 do Federal Information Processing Standard (FIPS) em algumas regiões. Para obter mais informações sobre a Norma federal de processamento de informações (Federal Information Processing Standard, FIPS) e endpoints da FIPS, consulte Visão geral da publicação 140-2 da FIPS . Para obter mais informações sobre endpoints de conexão do Git, consulte Regiões .	25 de outubro de 2018
CodeCommit adiciona três APIs	CodeCommit adicionou três APIs para suportar o trabalho com arquivos. Para obter mais informações sobre os endpoints de conexão do Git, consulte Permissões para ações em arquivos individuais e Referência de API do AWS CodeCommit .	27 de setembro de 2018

[CodeCommit notificação de histórico de documentação disponível por meio de feed RSS](#)

Agora você pode receber notificações sobre atualizações na CodeCommit documentação assinando um feed RSS.

29 de junho de 2018

Atualizações anteriores

A tabela a seguir descreve alterações importantes na documentação antes de 29 de junho de 2018.

Alteração	Descrição	Alterado em
Novo tópico	O tópico Limitar envios via push e mesclagens em ramificações foi adicionado. O tópico CodeCommit referência de permissões foi atualizado.	16 de maio de 2018
Nova seção	A seção Trabalhar com arquivos em repositórios do AWS CodeCommit foi adicionada. Os tópicos CodeCommit referência de permissões e Começando com AWS CodeCommit foram atualizados.	21 de fevereiro de 2018
Novo tópico	O tópico Configurar o acesso entre contas a um AWS CodeCommit repositório usando funções foi adicionado.	21 de fevereiro de 2018
Novo tópico	O tópico Integre AWS Cloud9 com AWS CodeCommit foi adicionado. O Integrações de produtos e serviços tópico foi atualizado com informações sobre AWS Cloud9.	1.º de dezembro de 2017
Nova seção	A seção Trabalhando com pull requests em AWS CodeCommit repositórios foi adicionada. A seção Autenticação e controle de acesso para AWS CodeCommit foi atualizada com informações sobre permissões para solicitações de pull e comentários. Ela também inclui instruções de política gerenciada atualizadas.	20 de novembro de 2017

Alteração	Descrição	Alterado em
Tópicos atualizados	O Integrações de produtos e serviços tópico foi atualizado para incluir links para clientes que desejam atualizar seus pipelines existentes para usar o Amazon CloudWatch Events para iniciar pipelines em resposta às mudanças em um CodeCommit repositório.	11 de outubro de 2017
Novos tópicos	A seção Autenticação e controle de acesso para AWS CodeCommit foi adicionada. Ela substitui o tópico Referência de permissões de acesso.	11 de setembro de 2017
Tópicos atualizados	A seção Gerenciar triggers para um repositório foi atualizada para refletir as alterações na configuração de triggers. Tópicos e imagens foram atualizados em todo o guia para refletir as alterações na barra de navegação.	29 de agosto de 2017
Novo tópico	O tópico Trabalhar com preferências do usuário foi adicionado. O tópico Visualizar detalhes de tags foi atualizado. Os Integrações de produtos e serviços tópicos foram atualizados com informações sobre a integração com a Amazon CloudWatch Events.	3 de agosto de 2017
Novos tópicos	Os Integre o Visual Studio com o AWS CodeCommit tópicos Integrar o Eclipse com AWS CodeCommit e foram adicionados.	29 de junho de 2017
Tópico atualizado	CodeCommit agora está disponível em duas regiões adicionais: Ásia-Pacífico (Mumbai) e Canadá (Central). O tópico Regiões e endpoints de conexão do Git foi atualizado.	29 de junho de 2017
Tópico atualizado	CodeCommit agora está disponível em quatro regiões adicionais: Ásia-Pacífico (Seul), América do Sul (São Paulo), Oeste dos EUA (Norte da Califórnia) e Europa (Londres). O tópico Regiões e endpoints de conexão do Git foi atualizado.	6 de junho de 2017

Alteração	Descrição	Alterado em
Tópico atualizado	CodeCommit agora está disponível em quatro regiões adicionais: Ásia-Pacífico (Tóquio), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney) e Europa (Frankfurt). O Regiões e endpoints de conexão do Git tópico foi atualizado para fornecer informações sobre os endpoints de conexão do Git e as regiões suportadas para CodeCommit.	25 de maio de 2017
Novo tópico	O tópico Comparar e mesclar ramificações foi adicionado. O conteúdo da seção Como trabalhar com ramificações foi atualizado com informações sobre o uso do console do CodeCommit para trabalhar com ramificações em um repositório.	18 de maio de 2017
Novo tópico	O tópico Comparar confirmações foi adicionado com informações sobre comparação de confirmações. A estrutura do guia do usuário foi atualizada para trabalhar com repositórios , confirmações e ramificações .	28 de março de 2017
Tópico atualizado	O tópico Visualizar detalhes de confirmação foi atualizado com informações sobre a visualização da diferença entre uma confirmação e o pai dela no console e o uso do comando get-differences para visualizar as diferenças entre confirmações usando a AWS CLI.	24 de janeiro de 2017
Novo tópico	O Registrando chamadas de AWS CodeCommit API com AWS CloudTrail tópico foi adicionado com informações sobre como registrar conexões e CodeCommit usar AWS CloudFormation.	11 de janeiro de 2017
Novo tópico	O Para usuários de HTTPS usando credenciais do Git tópico foi adicionado com informações sobre como configurar conexões para CodeCommit usar credenciais do Git por HTTPS.	22 de dezembro de 2016

Alteração	Descrição	Alterado em
Tópico atualizado	O Integrações de produtos e serviços tópico foi atualizado para incluir informações sobre a integração com AWS CodeBuild o.	5 de dezembro de 2016
Tópico atualizado	CodeCommit agora está disponível em outra região, Europa (Irlanda). O Regiões e endpoints de conexão do Git tópico foi atualizado para fornecer informações sobre os endpoints de conexão do Git e as regiões suportadas para. CodeCommit	16 de novembro de 2016
Tópico atualizado	CodeCommit agora está disponível em outra região, Oeste dos EUA (Oregon). O Regiões e endpoints de conexão do Git tópico foi atualizado para fornecer informações sobre os endpoints de conexão do Git e as regiões suportadas para. CodeCommit	14 de novembro de 2016
Novo tópico	O Criar um gatilho para uma função do Lambda tópico foi atualizado para refletir a capacidade de criar CodeCommit acionadores como parte da criação da função Lambda. Esse processo simplificado simplifica a criação do gatilho e configura automaticamente o gatilho com as permissões necessárias para invocar CodeCommit a função Lambda. O Criar um gatilho para uma função do Lambda existente tópico foi adicionado para incluir informações sobre a criação de acionadores para funções Lambda existentes no console. CodeCommit	19 de outubro de 2016
Novo tópico	CodeCommit agora está disponível em outra região, Leste dos EUA (Ohio). O Regiões e endpoints de conexão do Git tópico foi adicionado para fornecer informações sobre os endpoints de conexão do Git e as regiões suportadas para. CodeCommit	17 de outubro de 2016
Atualização de tópico	O Integrações de produtos e serviços tópico foi atualizado para incluir informações sobre a integração com AWS Elastic Beanstalk o.	13 de outubro de 2016

Alteração	Descrição	Alterado em
Atualização de tópico	O Integrações de produtos e serviços tópico foi atualizado para incluir informações sobre a integração com CloudFormation o.	6 de outubro de 2016
Atualização de tópico	O tópico Para conexões SSH no Windows foi revisado para fornecer orientações sobre o uso de um emulador de Bash para conexões SSH no Windows, ao invés da suíte de ferramentas PuTTY.	29 de setembro de 2016
Atualização de tópico	Os Começando com CodeCommit tópicos Visualizar detalhes de confirmação e foram atualizados para incluir informações sobre o Visualizador de Confirmações no CodeCommit console. O tópico Cotas foi atualizado com o aumento do número de referências permitido em um único push.	14 de setembro de 2016
Atualização de tópico	Os Começando com CodeCommit tópicos Visualizar detalhes de confirmação e foram atualizados para incluir informações sobre a visualização do histórico de confirmações no CodeCommit console.	28 de julho de 2016
Novos tópicos	Os tópicos Migrar um repositório Git para AWS CodeCommit e Migre conteúdo local ou não versionado para AWS CodeCommit foram adicionados.	29 de junho de 2016
Atualização de tópico	Atualizações menores foram feitas nos tópicos Solução de problemas e Para conexões HTTPS no Windows com o auxiliar de AWS CLI credenciais .	22 de junho de 2016
Atualização de tópico	Os tópicos Integrações de produtos e serviços e Referênci a de permissões de acesso foram atualizados para incluir informações sobre a integração com CodePipeline.	18 de abril de 2016

Alteração	Descrição	Alterado em
Novos tópicos	A seção Gerenciar triggers para um repositório foi adicionada. Os novos tópicos incluem exemplos, inclusive políticas e amostras de código, de como criar, editar e excluir triggers.	7 de março de 2016
Novo tópico	O tópico Integrações de produtos e serviços foi adicionado. Atualizações menores foram feitas em Solução de problemas .	7 de março de 2016
Atualização de tópico	Além da impressão digital do MD5 servidor, a impressão digital SHA256 do servidor CodeCommit foi adicionada ao e. Para conexões SSH no Linux, macOS ou Unix Para conexões SSH no Windows	9 de dezembro de 2015
Novo tópico	O tópico Procurar arquivos em um repositório foi adicionado. Novos problemas foram adicionados a Solução de problemas . Melhorias e correções menores foram feitas em todo o guia do usuário.	5 de outubro de 2015
Novo tópico	O tópico Para usuários de SSH que não usam o AWS CLI foi adicionado. Os tópicos na seção Configurando foram simplificados. Foi fornecida orientação para ajudar os usuários a determinarem quais etapas seguir para seus sistemas operacionais e protocolos preferidos.	5 de agosto de 2015
Atualização de tópico	Esclarecimentos e exemplos foram adicionados às etapas do ID de chave SSH em SSH e Linux, macOS ou Unix: configure as chaves públicas e privadas para Git e CodeCommit e Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit .	24 de julho de 2015
Atualização de tópico	As etapas em Etapa 3: configurar as chaves públicas e privadas para o Git e CodeCommit foram atualizadas para abordar um problema com o IAM e o salvamento do arquivo de chave pública.	22 de julho de 2015

Alteração	Descrição	Alterado em
Atualização de tópico	Solução de problemas foi atualizado com auxílios de navegação. Foram adicionadas mais informações sobre solução de problemas de conjuntos de chaves de credenciais.	20 de julho de 2015
Atualização de tópico	Mais informações sobre AWS Key Management Service permissões foram adicionadas aos tópicos AWS KMS e criptografia e à Referência de Permissões de Acesso.	17 de julho de 2015
Atualização de tópico	Outra seção foi adicionada a Solução de problemas com informações sobre solução de problemas com o AWS Key Management Service.	10 de julho de 2015
Lançamento inicial	Esta é a versão inicial do Guia do usuário do CodeCommit .	9 de julho de 2015

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.