

Guia do usuário

AWS CodeBuild



Versão da API 2016-10-06

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeBuild: Guia do usuário

Copyright © 2025 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 desprestige 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 é AWS CodeBuild?	1
.....	1
Como correr CodeBuild	1
Preços para CodeBuild	3
Como faço para começar com CodeBuild?	3
Conceitos	3
Como CodeBuild funciona	3
Próximas etapas	5
Conceitos básicos	6
Conceitos básicos que usam o console	6
Etapa 1: Criar o código-fonte	7
Etapa 2: Criar o arquivo buildspec	10
Etapa 3: Criar dois buckets do S3	12
Etapa 4: fazer upload do código-fonte e do arquivo buildspec	13
Etapa 5: criar o projeto de compilação	14
Etapa 6: executar a compilação	16
Etapa 7: visualizar informações resumidas da compilação	17
Etapa 8: visualizar informações detalhadas da compilação	18
Etapa 9: obter o artefato de saída da compilação	19
Etapa 10: Excluir os buckets do S3	20
Encapsulamento	20
Conceitos básicos do uso da AWS CLI	21
Etapa 1: Criar o código-fonte	22
Etapa 2: Criar o arquivo buildspec	24
Etapa 3: Criar dois buckets do S3	27
Etapa 4: fazer upload do código-fonte e do arquivo buildspec	27
Etapa 5: criar o projeto de compilação	29
Etapa 6: executar a compilação	33
Etapa 7: visualizar informações resumidas da compilação	34
Etapa 8: visualizar informações detalhadas da compilação	37
Etapa 9: obter o artefato de saída da compilação	40
Etapa 10: Excluir os buckets do S3	41
Encapsulamento	41
Exemplos baseados em caso de uso	42

Exemplos entre serviços	43
Exemplo do Amazon ECR	44
Exemplo do Amazon EFS	51
AWS CodePipeline amostras	57
AWS Config amostra	68
Exemplo de notificações de compilação	70
Exemplo de emblemas de compilação	85
Criar um projeto de compilação com emblemas de compilação	86
Acesse os emblemas de AWS CodeBuild construção	89
Publique emblemas de CodeBuild construção	90
CodeBuild status do crachá	90
Exemplo de relatório de teste	90
Executar o exemplo de relatório de teste	91
Amostras do Docker para CodeBuild	97
Docker no exemplo de imagem personalizada	98
Exemplo de servidor de criação de imagens do Docker	101
Exemplo de compilações do Docker do Windows	104
Exemplo “Publicar imagem do Docker no Amazon ECR”	106
Registro privado com AWS Secrets Manager amostra	115
Saída de compilação hospedada em um bucket do S3	119
Exemplo de várias entradas e saídas	123
Criar um projeto de compilação com várias entradas e saídas	124
Criar um projeto sem uma origem	127
Versões do tempo de execução no exemplo de arquivo buildspec	127
Atualizar a versão do runtime no arquivo buildspec	128
Especificar dois tempos de execução	132
Exemplo de versão de fonte	136
Especifique uma versão GitHub do repositório com um ID de confirmação	138
Especifique uma versão do GitHub repositório com uma referência e um ID de confirmação	140
Exemplos de repositórios de origem de terceiros	141
Executar o exemplo do Bitbucket	141
Execute a amostra do GitHub Enterprise Server	147
Execute a solicitação GitHub pull e a amostra do filtro de webhook	153
Tutorial: Assinatura de código da Apple com Fastlane ao CodeBuild usar o S3 para armazenamento de certificados	158

Tutorial: Assinatura de código da Apple com o Fastlane em CodeBuild Usando GitHub para armazenamento de certificados	164
Defina os nomes dos artefatos no momento da compilação	170
Executar exemplos do Windows	173
Executar os exemplos do Windows	173
Estrutura de diretório	175
F# e o .NET Framework	175
Visual Basic e o .NET Framework	175
Arquivos	175
F# e o .NET Framework	175
Visual Basic e o .NET Framework	180
Planejar uma compilação	194
Referência de buildspec	197
Nome do arquivo buildspec e local de armazenamento	197
Sintaxe de buildspec	198
version	201
run-as	201
env	202
proxy	207
phases	207
relatórios	211
Artefatos	213
cache	220
Exemplo de buildspec	222
Versões de buildspec	225
Referência de buildspec em lote	225
lote	226
batch/build-graph	226
batch/build-list	229
batch/build-matrix	232
batch/build-fanout	234
Referência de ambiente de compilação	236
Imagens do Docker fornecidas por CodeBuild	237
Obter a lista das imagens do Docker atuais	237
EC2 imagens de computação	238
Imagens de computação do Lambda	240

Imagens obsoletas CodeBuild	244
Runtimes disponíveis	246
Versões do tempo de execução	265
Modos e tipos de computação do ambiente de compilação	270
Sobre computação	270
Sobre os tipos de ambiente de capacidade reservada	271
Sobre os tipos de ambiente sob demanda	324
Shells e comandos em ambientes de compilação	336
Variáveis de ambiente em ambientes de compilação	337
Tarefas em segundo plano em ambientes de compilação	343
Projetos de compilação	345
Criar um projeto de compilação	345
Pré-requisitos	346
Criar um projeto de compilação (console)	346
Criar um projeto de compilação (AWS CLI)	368
Criar um projeto de compilação (AWS SDKs)	389
Criar um projeto de compilação (AWS CloudFormation)	389
Criar uma regra de notificação	389
Alterar as configurações do projeto de compilação	393
Alterar configurações de um projeto de compilação (console)	393
Alterar configurações de um projeto de compilação (AWS CLI)	418
Alterar as configurações de um projeto de construção (AWS SDKs)	419
Vários tokens de acesso	419
Etapa 1: Criar um segredo ou uma CodeConnections conexão do Secrets Manager	420
Etapa 2: conceder acesso à função IAM do CodeBuild projeto aos segredos do Secrets Manager	420
Etapa 3: Configurar o Secrets Manager ou CodeConnections tokens	422
Opções de configuração adicionais	426
Excluir projetos de compilação	429
Excluir um projeto de compilação (console)	430
Excluir um projeto de compilação (AWS CLI)	430
Excluir um projeto de compilação (AWS SDKs)	431
Obtenha um projeto de construção público URLs	431
Compartilhar projetos de compilação	432
Compartilhar um projeto	433
Serviços relacionados	436

Acessar projetos compartilhados	436
Cancelar o compartilhamento de um projeto compartilhado	436
Identificar um projeto compartilhado	437
Permissões para projetos compartilhados	437
Projetos de compilação de tags	438
Adicionar uma tag a um projeto	439
Visualizar tags de um projeto	440
Editar tags de um projeto	441
Remover uma tag de um projeto	442
Usar executores	443
GitHub Ações	444
GitLab corredores	466
Corredor Buildkite	481
Usar webhooks	503
Práticas recomendadas para usar webhooks	504
Filtrar eventos de webhook do Bitbucket	505
GitHub webhooks globais e organizacionais	518
GitHub webhooks manuais	525
GitHub eventos de webhook	527
GitLab webhooks em grupo	543
GitLab webhooks manuais	549
GitLab eventos de webhook	550
Webhooks manuais do Buildkite	565
Exibir os detalhes do projeto de compilação	566
Visualizar detalhes de um projeto de compilação (console)	567
Visualizar detalhes de um projeto de compilação (AWS CLI)	567
Exibir detalhes de um projeto de construção (AWS SDKs)	569
Exibir os nomes do projeto de compilação	570
Visualizar uma lista de nomes de projetos de compilação (console)	570
Visualizar uma lista de nomes de projetos de compilação (AWS CLI)	570
Veja uma lista de nomes de projetos de construção (AWS SDKs)	572
Compilações	573
Executar compilações manualmente	574
Executar uma compilação localmente	575
Executar uma compilação (console)	578
Executar uma compilação (AWS CLI)	579

Executar uma compilação em lote (AWS CLI)	586
Começar a executar compilações automaticamente (AWS CLI)	588
Parar de executar compilações automaticamente (AWS CLI)	589
Executar uma compilação (AWS SDKs)	589
Execute compilações na computação Lambda	590
Quais ferramentas e runtimes serão incluídos nas imagens do Docker do ambiente de runtime selecionado que são executadas no AWS Lambda?	590
E se a imagem selecionada não incluir as ferramentas de que preciso?	590
Em quais regiões oferecem suporte à AWS Lambda computação? CodeBuild	591
Limitações da AWS Lambda computação	591
Implemente uma função Lambda usando o AWS SAM CodeBuild Lambda Java	592
Crie um aplicativo React de página única com CodeBuild Lambda Node.js	596
Atualize a configuração de uma função Lambda com o CodeBuild Lambda Python	599
Executar compilações em frotas de capacidade reservada	604
Criar uma frota de capacidade reservada	605
Práticas recomendadas	607
Posso compartilhar uma frota de capacidade reservada em vários CodeBuild projetos?	607
Como funciona a computação baseada em atributos?	608
Posso especificar manualmente uma EC2 instância da Amazon para minha frota?	608
Quais regiões são compatíveis com frotas de capacidade reservada?	608
Como configurar uma frota macOS de capacidade reservada?	609
Como configuro uma Amazon Machine Image (AMI) personalizada para uma frota de capacidade reservada?	610
Limitações de frotas de capacidade reservada	612
Propriedades da frota de capacidade reservada	612
Amostras de capacidade reservada	617
Executar compilações em lote	619
Perfil de segurança	619
Tipos de compilação em lote	619
Modo de relatório em lote	624
Mais informações	624
Execute testes paralelos	624
Support em AWS CodeBuild	626
Habilite a execução paralela de testes em compilações em lote	628
Use o comando <code>codebuild-tests-run</code> CLI	629
Use o comando <code>codebuild-glob-search</code> CLI	632

Sobre a divisão de testes	634
Mescle automaticamente relatórios de criação individuais	635
Amostras de execução paralela de testes	637
Compilações em cache	648
Armazenamento em cache do Amazon S3	648
Armazenamento em cache local	655
Especificar um cache local	657
Depurar compilações	659
Depure compilações com sandbox CodeBuild	659
Depurar compilações com o Session Manager	660
Depure compilações com sandbox CodeBuild	660
Depurar compilações com o Gerenciador de Sessões	690
Excluir compilações	695
Excluir compilações (AWS CLI)	695
Excluir compilações (AWS SDKs)	696
Repetir compilações manualmente	696
Repetir uma compilação manualmente (console)	697
Repetir uma compilação manualmente (AWS CLI)	697
Repetir uma compilação manualmente (AWS SDKs)	698
Repetir compilações automaticamente	698
Repetir uma compilação automaticamente (console)	698
Repetir uma compilação automaticamente (AWS CLI)	699
Repetir automaticamente um build (AWS SDKs)	699
Interromper compilações	700
Parar uma compilação (console)	700
Parar uma compilação (AWS CLI)	701
Parar uma compilação (AWS SDKs)	701
Interromper compilações em lote	701
Interromper uma compilação (console)	702
Interromper uma compilação em lote (AWS CLI)	702
Interromper uma compilação em lote (AWS SDKs)	703
O gatilho é criado automaticamente	703
Criar triggers de compilação	703
Editar triggers de compilação	707
Visualizar detalhes de compilação	710
Visualizar detalhes de compilação (console)	710

Visualizar detalhes de compilação (AWS CLI)	711
Visualizar detalhes de compilação (AWS SDKs)	712
Transições de fase de compilação	712
Exibir compilação IDs	712
Exibir uma lista de IDs compilações (console)	712
Veja uma lista de build IDs (AWS CLI)	713
Exibir uma lista de compilações em lote IDs (AWS CLI)	714
Veja uma lista de build IDs (AWS SDKs)	716
Exibir compilação IDs para um projeto de compilação	716
Exibir uma lista de compilações IDs para um projeto de compilação (console)	716
Exibir uma lista de IDs compilações para um projeto de construção (AWS CLI)	717
Exibir uma lista de compilações em lote IDs para um projeto de compilação (AWS CLI)	718
Exibir uma lista de IDs compilações para um projeto de construção (AWS SDKs)	720
Relatórios de teste	721
Criar relatórios de teste	722
Criar relatórios de cobertura de código	723
.....	723
Criar um relatório de cobertura de código	724
Relatórios de descoberta automática	725
Configurar a descoberta automática de relatórios usando o console	727
Configure a descoberta automática de relatórios usando variáveis de ambiente do projeto .	727
Grupos de relatórios	728
Criar um grupo de relatórios	729
Nomenclatura do grupo de relatórios	735
Compartilhar grupos de relatórios	736
Especificar arquivos de teste	741
Especificar comandos de teste	742
Marcar um grupo de relatórios	742
Atualizar um grupo de relatórios	749
Framework de teste	752
Configurar o Jasmine	752
Configurar o Jest	755
Configurar o pytest	756
Configurar RSpec	757
Visualizar relatórios de teste	758
Visualizar relatórios de teste para uma compilação	759

Visualizar relatórios de teste para um grupo de relatórios	759
Visualizar relatórios de teste em sua conta da AWS	759
Permissões de relatório de teste	760
Perfil do IAM para relatórios de testes	760
Permissões para operações de relatórios de testes	762
Exemplos de permissões de relatórios de testes	762
Status do relatório de teste	763
Suporte à VPC	765
Casos de uso	765
Melhores práticas para VPCs	766
Limitações do VPCs	767
Permita o acesso da Amazon VPC em seus projetos CodeBuild	767
Solucionar problemas de configuração da VPC	768
Usar VPC endpoints	769
Antes de criar VPC endpoints	769
Crie endpoints VPC para CodeBuild	770
Crie uma política de VPC endpoint para CodeBuild	770
Use um servidor proxy CodeBuild gerenciado	771
Definir uma configuração de proxy gerenciado para frotas de capacidade reservada	772
Administre uma frota de capacidade CodeBuild reservada	773
Usar um servidor de proxy	773
Configurar componentes necessários para execução CodeBuild em um servidor proxy	774
Execute CodeBuild em um servidor proxy explícito	777
Execute CodeBuild em um servidor proxy transparente	782
Executar um gerenciador de pacotes e outras ferramentas em um servidor de proxy	783
AWS CloudFormation Modelo de VPC	785
Registro em log e monitoramento	792
Registrar chamadas de CodeBuild API	792
Sobre AWS CodeBuild as informações em CloudTrail	792
Sobre as entradas do arquivo de AWS CodeBuild log	793
Monitorar compilações	796
CloudWatch métricas	797
CloudWatch métricas de utilização de recursos	799
CloudWatch dimensões	801
CloudWatch alarmes	801
Exibir CodeBuild métricas	802

Exibir CodeBuild métricas de utilização de recursos	804
Crie CodeBuild alarmes em CloudWatch	808
Segurança	810
Proteção de dados	810
Criptografia de dados	812
Gerenciamento de chaves	813
Privacidade do tráfego	813
Gerenciamento de identidade e acesso	814
Visão geral do gerenciamento de acesso	814
Usar políticas baseadas em identidade	818
AWS CodeBuild referência de permissões	851
Usar tags para controlar o acesso aos recursos do AWS CodeBuild	858
Visualizar recursos no console	862
Validação de conformidade	862
Resiliência	863
Segurança da infraestrutura	864
Acesso do provedor de origem	864
Criar e armazenar um token em um segredo do Secrets Manager	865
GitHub e acesso ao GitHub Enterprise Server	868
Acesso ao Bitbucket	880
GitLab acesso	888
Prevenção contra o ataque do “substituto confuso” em todos os serviços	895
Tópicos avançados	898
Permita que os usuários interajam com CodeBuild	898
CodeBuild Permitir interagir com outros AWS serviços	905
Criptografar saídas da compilação	913
Interaja com CodeBuild o uso do AWS CLI	916
Referência da linha de comando	917
AWS SDKs e referência de ferramentas	918
Suporte AWS SDKs e ferramentas para AWS CodeBuild	918
Trabalhando com AWS SDKs	919
Especifique o CodeBuild endpoint	920
Especifique o AWS CodeBuild endpoint (AWS CLI)	921
Especifique o AWS CodeBuild endpoint (AWS SDK)	921
Use CodeBuild com CodePipeline	924
Pré-requisitos	925

Criar um pipeline (console)	927
Criar um pipeline (AWS CLI)	931
Ação Adicionar uma compilação	936
Ação Adicionar um teste	940
Use CodeBuild com Codecov	943
Integrar o Codecov em um projeto de compilação	944
Use CodeBuild com Jenkins	947
Configurar o Jenkins	947
Instalar o plug-in	948
Usar o plug-in	948
Use CodeBuild com aplicativos sem servidor	950
Recursos relacionados	950
Avisos de terceiros	950
1) Imagem do Docker de base: windowsservercore	951
2) Imagem do Docker baseada em Windows: choco	952
3) Imagem do Docker baseada em Windows: git – versão 2.16.2	953
4) imagem Docker baseada em Windows— --versão 15.0.26320.2 microsoft-build-tools	953
5) Imagem do Docker baseada em Windows: nuget.commandline – versão 4.5.1	957
7) Imagem do Docker baseada em Windows: netfx-4.6.2-devpack	957
8) Imagem do Docker baseada em Windows: visualfsharp tools, v 4.0	959
9) imagem Docker baseada em janelas — -4.6 netfx-pcl-reference-assemblies	959
10) Imagem do Docker baseada em Windows: visualcppbuildtools v 14.0.25420.1	963
11) Imagem do Docker baseada em Windows — 3-ondemand-package.cab microsoft- windows-netfx	967
12) Imagem do Docker baseada em Windows: dotnet-sdk	968
Use chaves de CodeBuild condição como variáveis de função de serviço do IAM	969
AWS CodeBuild chaves de condição	969
Aplique as configurações de conectividade VPC em seus projetos e frotas	970
Evite modificações não autorizadas no buildspec do projeto	971
Restrinja os tipos de computação para suas compilações	971
Configurações variáveis do ambiente de controle	972
Use variáveis em nomes de chaves de condição	973
Verifique a existência de atributos nas solicitações de API	974
Exemplos de código	975
Conceitos básicos	975
Ações	976

Solução de problemas	993
Artefatos de referência de compilações Apache Maven do repositório errado	994
Comandos de compilação executados como raiz por padrão	996
As compilações podem falhar quando nomes de arquivos têm caracteres que não sejam do inglês.	996
As compilações podem falhar ao obter parâmetros do Amazon EC2 Parameter Store	997
Não é possível acessar o filtro da ramificação de acesso no console do CodeBuild	998
Não é possível visualizar o êxito ou a falha de compilação	998
Status de compilação não comunicado ao provedor de origem	999
Não é possível localizar e selecionar a imagem de base da plataforma Windows Server Core 2019.	999
Comandos anteriores em arquivos buildspec não são reconhecidos por comandos posteriores	1000
Erro: "acesso negado" ao tentar fazer download do cache	1000
Erro: "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" ao usar uma imagem de compilação personalizada	1001
Erro: "O contêiner de compilação foi encontrado inativo antes de concluir a compilação. O contêiner de compilação morreu porque estava sem memória ou a imagem do Docker não é suportada. ErrorCode: 500"	1002
Erro: "Cannot connect to the Docker daemon (Não é possível conectar-se ao daemon do Docker)" ao executar uma compilação	1002
Erro: "não CodeBuild está autorizado a executar: sts:AssumeRole" ao criar ou atualizar um projeto de compilação	1004
Erro: "Erro ao chamar GetBucketAcl: ou o proprietário do bucket mudou ou a função de serviço não tem mais permissão para chamar s3:GetBucketAcl"	1005
Erro: "Failed to upload artifacts: Invalid arn (Falha ao fazer upload de artefatos: arn inválido)" ao executar uma compilação	1005
Erro: "falha do clone do Git: não é possível acessar 'your-repository-URL': problema de certificado SSL: certificado autoassinado"	1005
Erro: "The bucket you are attempting to access must be addressed using the specified endpoint (O bucket que você está tentando acessar deve ser endereçado usando o endpoint especificado)" ao executar uma compilação	1006
Erro: "This build image requires selecting at least one runtime version" (Esta imagem de compilação requer a seleção de pelo menos um tempo de execução)	1006
Erro: "QUEUED: INSUFFICIENT_SUBNET" quando ocorre uma falha em uma compilação em uma fila de compilação	1007

Erro: "Não foi possível baixar o cache: RequestError: Falha na solicitação de envio causada por: x509: Falha ao carregar raízes do sistema e nenhuma raiz fornecida"	1008
Erro: "Não foi possível baixar o certificado do S3. AccessDenied"	1008
Erro: "não foi possível localizar as credenciais"	1009
RequestError erro de tempo limite ao executar CodeBuild em um servidor proxy	1010
O bourne shell (sh) deve existir em imagens de compilação	1012
Aviso: "ignorando a instalação de tempos de execução. A seleção de versão de tempo de execução não é compatível com esta imagem de compilação" ao executar uma compilação .	1012
Erro: "Não foi possível verificar a JobWorker identidade"	1012
Falha ao iniciar a compilação	1013
Acessando GitHub metadados em compilações armazenadas em cache localmente	1013
AccessDenied: O proprietário do bucket do grupo de relatórios não corresponde ao proprietário do bucket do S3...	1013
Erro: "Suas credenciais não têm um ou mais escopos de privilégios necessários" ao criar um projeto com CodeBuild CodeConnections	1014
Erro: "Desculpe, nenhum terminal foi solicitado - não foi possível obter a entrada" ao compilar com o comando de instalação do Ubuntu	1015
Cotas	1017
Cotas de serviço	1017
Outros limites	1023
Projetos de compilação	1023
Compilações	1024
Frotas de computação	1024
Relatórios	1026
Tags	1026
Histórico de documentos	1028
Atualizações anteriores	1053
.....	mlxvii

O que é AWS CodeBuild?

AWS CodeBuild é um serviço de criação totalmente gerenciado na nuvem. CodeBuild compila seu código-fonte, executa testes de unidade e produz artefatos prontos para serem implantados. CodeBuild elimina a necessidade de provisionar, gerenciar e escalar seus próprios servidores de compilação. Ele fornece ambientes de compilação com pacotes predefinidos para linguagens populares de programação e ferramentas de compilação, como Apache Maven, Gradle, entre outras. Você também pode personalizar ambientes de compilação CodeBuild para usar suas próprias ferramentas de compilação. CodeBuild escala automaticamente para atender às solicitações de pico de construção.

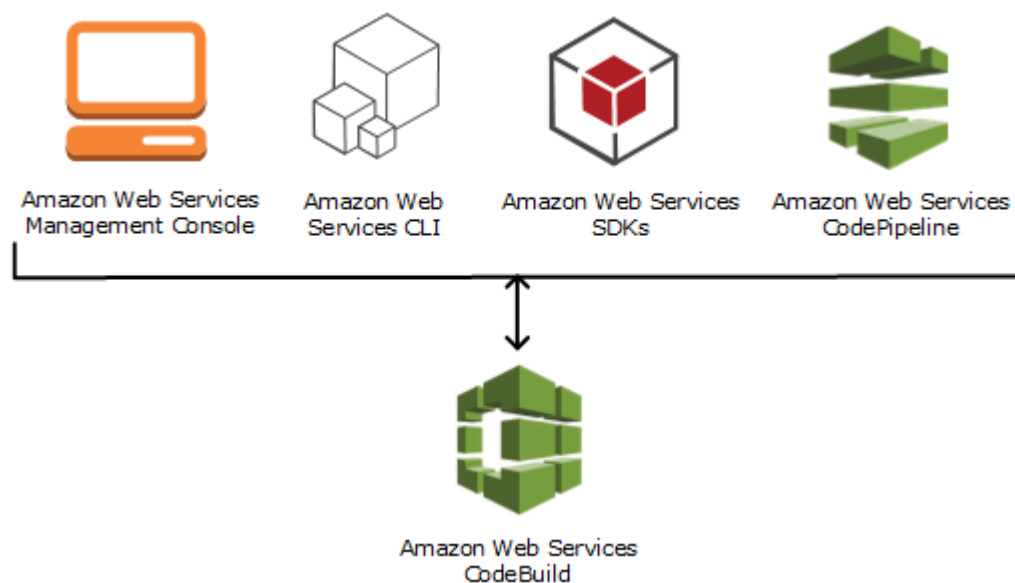
CodeBuild fornece os seguintes benefícios:

- Totalmente gerenciado — CodeBuild elimina a necessidade de configurar, corrigir, atualizar e gerenciar seus próprios servidores de compilação.
- Sob demanda — CodeBuild escala sob demanda para atender às suas necessidades de construção. Você paga somente pela quantidade de minutos de compilação que consumir.
- Pronto para uso — CodeBuild fornece ambientes de construção pré-configurados para as linguagens de programação mais populares. Tudo o que você precisa fazer é apontar para o seu script de compilação para iniciar sua primeira compilação.

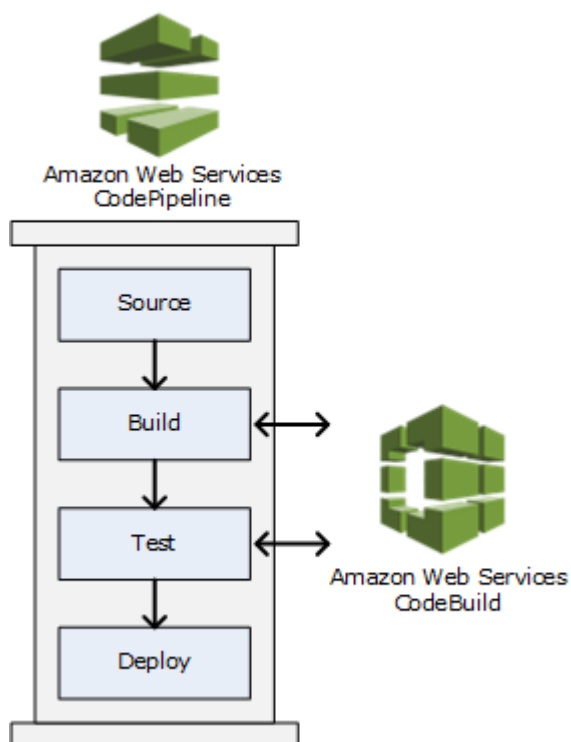
Para obter mais informações, consulte [AWS CodeBuild](#).

Como correr CodeBuild

Você pode usar o AWS CodePipeline console AWS CodeBuild ou para executar CodeBuild. Você também pode automatizar a execução do CodeBuild usando o AWS Command Line Interface (AWS CLI) ou o AWS SDKs



Como mostra o diagrama a seguir, você pode adicionar CodeBuild como uma ação de construção ou teste ao estágio de construção ou teste de um pipeline em AWS CodePipeline. AWS CodePipeline é um serviço de entrega contínua que você pode usar para modelar, visualizar e automatizar as etapas necessárias para liberar seu código. Isso inclui a compilação de seu código. Um pipeline é uma construção de fluxo de trabalho que descreve como as alterações de código atravessam um processo de lançamento.



Para usar CodePipeline para criar um pipeline e depois adicionar uma ação de CodeBuild criação ou teste, consulte [Use CodeBuild com CodePipeline](#). Para obter mais informações sobre CodePipeline, consulte o [Guia AWS CodePipeline do usuário](#).

O CodeBuild 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 insira 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](#).

Preços para CodeBuild

Para obter mais informações, consulte [Definição de preço do CodeBuild](#).

Como faço para começar com CodeBuild?

É recomendável que você realize as etapas a seguir:

1. Saiba mais CodeBuild lendo as informações em [Conceitos](#).
2. Experimente CodeBuild um cenário de exemplo seguindo as instruções em [Conceitos básicos que usam o console](#).
3. Use CodeBuild em seus próprios cenários seguindo as instruções em [Planejar uma compilação](#).

AWS CodeBuild conceitos

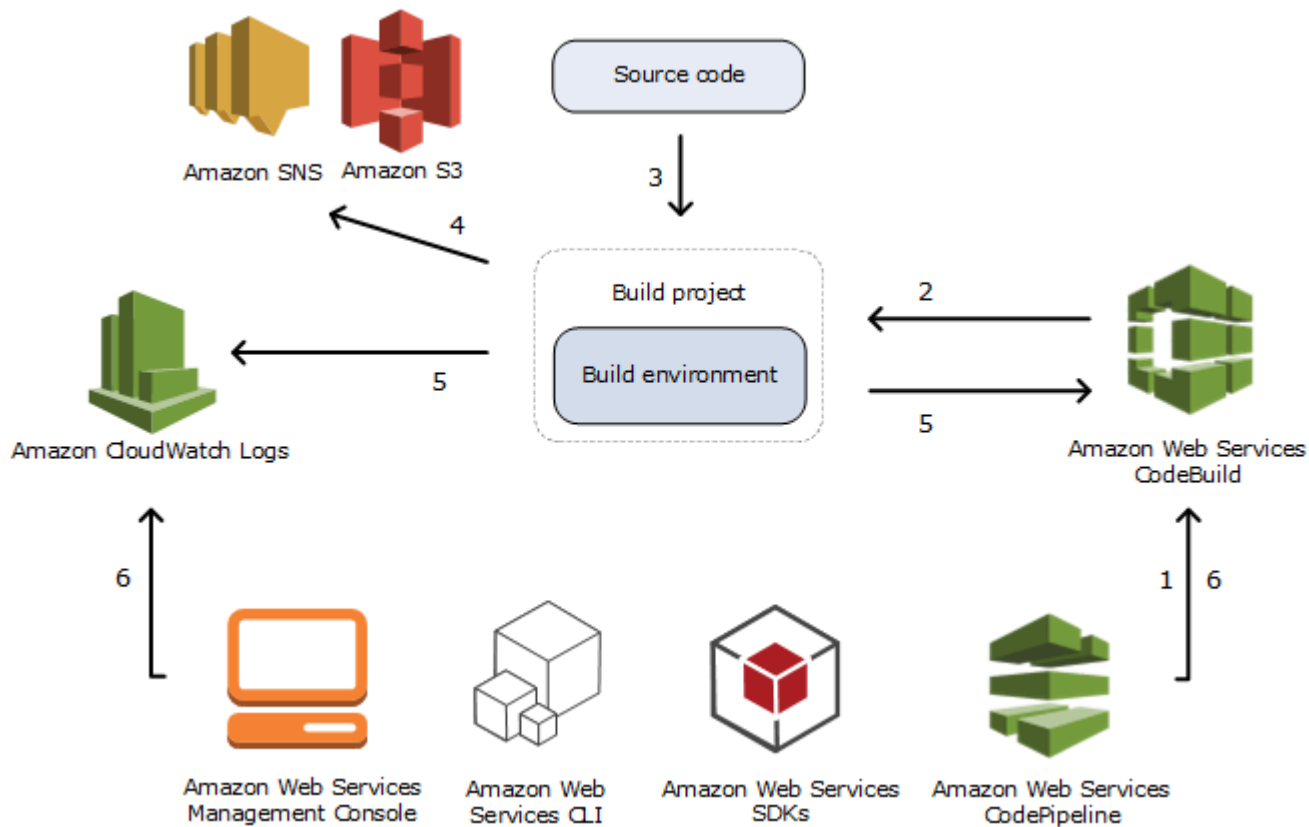
Os conceitos a seguir são importantes para entender como CodeBuild funciona.

Tópicos

- [Como CodeBuild funciona](#)
- [Próximas etapas](#)

Como CodeBuild funciona

O diagrama a seguir mostra o que acontece quando você executa uma compilação com CodeBuild:



1. Como entrada, você deve CodeBuild fornecer um projeto de construção. Um projeto de compilação inclui informações sobre como executar uma compilação, incluindo onde obter o código-fonte, qual ambiente de compilação usar, quais comandos de compilação executar e onde armazenar a saída da compilação. Um ambiente de compilação representa uma combinação de sistema operacional, tempo de execução da linguagem de programação e ferramentas CodeBuild usadas para executar uma compilação. Para obter mais informações, consulte:
 - [Criar um projeto de compilação](#)
 - [Referência de ambiente de compilação](#)
2. CodeBuild usa o projeto de construção para criar o ambiente de construção.
3. CodeBuild baixa o código-fonte no ambiente de compilação e, em seguida, usa a especificação de compilação (buildspec), conforme definida no projeto de compilação ou incluída diretamente no código-fonte. Um buildspec é uma coleção de comandos de compilação e configurações relacionadas, no formato YAML, CodeBuild usados para executar uma compilação. Para obter mais informações, consulte [Referência de buildspec](#).

4. Se houver alguma saída de compilação, o ambiente de compilação fará upload de sua saída para um bucket do S3. O ambiente de compilação também pode executar tarefas especificadas no buildspec (por exemplo, enviar notificações de compilação a um tópico do Amazon SNS). Para obter um exemplo, consulte [Exemplo de notificações de compilação](#).
5. Enquanto a compilação está em execução, o ambiente de compilação envia informações para CodeBuild a Amazon CloudWatch Logs.
6. Enquanto a compilação estiver em execução, você pode usar o AWS CodeBuild console ou AWS SDKs obter informações resumidas CodeBuild e detalhadas da compilação do Amazon CloudWatch Logs. AWS CLI Se você usa AWS CodePipeline para executar compilações, pode obter informações limitadas de CodePipeline compilação em.

Próximas etapas

Agora que você sabe mais sobre isso AWS CodeBuild, recomendamos as próximas etapas:

1. Experimente CodeBuild em um cenário de exemplo seguindo as instruções em [Conceitos básicos que usam o console](#).
2. Use CodeBuild em seus próprios cenários seguindo as instruções em [Planejar uma compilação](#).

Começando com CodeBuild

Nos tutoriais a seguir, você usa AWS CodeBuild para criar uma coleção de exemplos de arquivos de entrada de código-fonte em uma versão implantável do código-fonte.

Ambos os tutoriais têm a mesma entrada e resultados, mas um usa o AWS CodeBuild console e o outro usa o AWS CLI

Important

Não recomendamos que você use sua conta AWS root para concluir este tutorial.

Tópicos

- [Começando a AWS CodeBuild usar o console](#)
- [Começando a AWS CodeBuild usar o AWS CLI](#)

Começando a AWS CodeBuild usar o console

Neste tutorial, você usa AWS CodeBuild para criar uma coleção de exemplos de arquivos de entrada de código-fonte (criar artefatos de entrada ou criar entrada) em uma versão implantável do código-fonte (criar artefato de saída ou criar saída). Especificamente, você instrui CodeBuild a usar o Apache Maven, uma ferramenta de construção comum, para criar um conjunto de arquivos de classe Java em um arquivo Java Archive (JAR). Você não precisa estar familiarizado com o Apache Maven ou com o Java para utilizar este tutorial.

Você pode trabalhar com CodeBuild ele por meio do CodeBuild console AWS CodePipeline, AWS CLI, do ou do AWS SDKs. Este tutorial demonstra como usar o CodeBuild console. Para obter informações sobre como utilizar o CodePipeline, consulte [Use CodeBuild com CodePipeline](#).

Important

As etapas deste tutorial exigem que você crie recursos (por exemplo, um bucket do S3) que podem resultar em cobranças na sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados ao Amazon S3 e CloudWatch aos AWS

KMS registros. Para obter mais informações, consulte [AWS CodeBuild preços, preços](#) do [Amazon S3, preços](#) e [AWS Key Management Service preços](#) da [Amazon CloudWatch](#).

Tópicos

- [Etapa 1: Criar o código-fonte](#)
- [Etapa 2: Criar o arquivo buildspec](#)
- [Etapa 3: Criar dois buckets do S3](#)
- [Etapa 4: fazer upload do código-fonte e do arquivo buildspec](#)
- [Etapa 5: criar o projeto de compilação](#)
- [Etapa 6: executar a compilação](#)
- [Etapa 7: visualizar informações resumidas da compilação](#)
- [Etapa 8: visualizar informações detalhadas da compilação](#)
- [Etapa 9: obter o artefato de saída da compilação](#)
- [Etapa 10: Excluir os buckets do S3](#)
- [Encapsulamento](#)

Etapa 1: Criar o código-fonte

(Parte de: [Começando a AWS CodeBuild usar o console](#))

Nesta etapa, você cria o código-fonte que CodeBuild deseja criar no bucket de saída. O código-fonte consiste em dois arquivos classe Java e um arquivo Apache Maven Project Object Model (POM).

1. Em um diretório vazio em seu computador ou instância local, crie essa estrutura de diretório.

```
(root directory name)
|-- src
|   |-- main
|       |-- java
|   |-- test
|       |-- java
```

2. Use um editor de texto de sua preferência, crie esse arquivo, nomeie-o como `MessageUtil.java` e salve-o no diretório `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }

    public String salutationMessage() {
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

Esse arquivo classe cria como a saída o string de caracteres para ele transferido. O construtor MessageUtil define o string de caracteres. O método printMessage cria a saída. O método salutationMessage gera a saída Hi! seguida da string de caracteres.

3. Crie esse arquivo, nomeie-o TestMessageUtil.java e o salve no diretório /src/test/java.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
```

```
public void testSalutationMessage() {
    System.out.println("Inside testSalutationMessage()");
    message = "Hi!" + "Robert";
    assertEquals(message,messageUtil.salutationMessage());
}
}
```

Esse arquivo classe define a variável `message` na classe `MessageUtil` para `Robert`. O arquivo avalia, então, se a variável `message` foi definida com sucesso, verificando se as strings `Robert` e `Hi!Robert` aparecem na saída.

4. Crie esse arquivo, nomeie-o `pom.xml` e salve-o no diretório raiz (de nível superior).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>messageUtil</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
    <name>Message Utility Java Sample App</name>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
            </plugin>
        </plugins>
    </build>
</project>
```

O Apache Maven usa as instruções nesse arquivo para converter os arquivos `MessageUtil.java` e `TestMessageUtil.java` em um arquivo denominado `messageUtil-1.0.jar` e, em seguida, executar os testes especificados.

Neste momento, a estrutura de diretório deve se parecer como essa.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Etapa 2: Criar o arquivo buildspec

(Etapa anterior: [Etapa 1: Criar o código-fonte](#))

Nesta etapa, você cria um arquivo de especificação de build (build spec). Um buildspec é uma coleção de comandos de compilação e configurações relacionadas, no formato YAML, CodeBuild usados para executar uma compilação. Sem uma especificação de compilação, CodeBuild não é possível converter com êxito sua entrada de compilação em saída de compilação ou localizar o artefato de saída de compilação no ambiente de compilação para fazer upload em seu bucket de saída.

Crie esse arquivo, nomeie-o `buildspec.yml` e salve-o no diretório raiz (de nível superior).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
```

```
commands:
  - echo Build started on `date`
  - mvn install
post_build:
  commands:
    - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Como a declaração de build spec deve corresponder ao YAML, o espaçamento na declaração de build spec é importante. Se o número de espaços em sua declaração de build spec não corresponder a essa, o build pode falhar imediatamente. Você pode usar um validador YAML para testar se a declaração de build spec é válida para o YAML.

Note

Em vez de incluir um arquivo de build spec em seu código-fonte, você pode declarar comandos de build separadamente ao criar um projeto de build. Isso é útil no caso de você compilar seu código-fonte com diferentes comandos de build sem atualizar o repositório de código fonte a cada vez. Para obter mais informações, consulte [Sintaxe de buildspec](#).

Nesta declaração de build spec:

- `version` representa a versão do build spec padrão em uso. Essa declaração de build spec usa a última versão, `0.2`.
- `phases` representa as fases do build durante as quais você pode instruir o CodeBuild para executar comandos. Essas fases de build são aqui listadas como `install`, `pre_build`, `build` e `post_build`. Não é possível alterar os nomes das fases de build, nem criar nomes adicionais.

Neste exemplo, durante a `build` fase, CodeBuild executa o `mvn install` comando. Esse comando instrui o Apache Maven a compilar, testar e empacotar os arquivos classe Java em um artefato de saída do build. Para completar, alguns comandos `echo` são incluídos em cada fase de build, neste exemplo. Quando você visualizar informações de compilação detalhadas posteriormente neste tutorial, a saída desses comandos `echo` poderá ajudar a compreender

melhor como o CodeBuild executa comandos e em que ordem. (Embora todas as fases da compilação estejam incluídas neste exemplo, não é necessário incluir uma fase da compilação se você não pretende executar comandos durante essa fase.) Para cada fase de compilação, CodeBuild executa cada comando especificado, um por vez, na ordem listada, do início ao fim.

- `artifacts` representa o conjunto de artefatos de saída de compilação que são CodeBuild carregados no bucket de saída. `files` representa os arquivos a serem incluídos na saída da compilação. CodeBuild carrega o único `messageUtil-1.0.jar` arquivo encontrado no diretório `target` relativo no ambiente de compilação. O nome de arquivo `messageUtil-1.0.jar` e o nome de diretório `target` são baseados na maneira como o Apache Maven cria e armazena artefatos de saída do build somente para este exemplo. Em seus próprios builds, esses nomes de arquivos e diretórios são diferentes.

Para obter mais informações, consulte o [Referência de buildspec](#).

Neste momento, a estrutura de diretório deve se parecer como essa.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Etapa 3: Criar dois buckets do S3

(Etapa anterior: [Etapa 2: Criar o arquivo buildspec](#))

Embora seja possível usar um único bucket para este tutorial, o uso de dois buckets facilita a visualização da origem da entrada da compilação e do destino da saída da compilação.

- Um desses buckets (o bucket de entrada) armazena a entrada da compilação. Neste tutorial, o nome desse bucket de entrada é `codebuild-region-ID-account-ID-input-bucket`, onde *region-ID* é a AWS região do bucket e *account-ID* é o ID da sua AWS conta.
- O outro bucket (o bucket de saída) armazena a saída da compilação. Neste tutorial, o nome deste bucket de saída é `codebuild-region-ID-account-ID-output-bucket`.

Se você escolheu nomes diferentes para esses buckets, certifique-se de usá-los ao longo deste tutorial.

Esses dois buckets devem estar na mesma AWS região das suas compilações. Por exemplo, se você CodeBuild instruir a execução de uma compilação na região Leste dos EUA (Ohio), esses buckets também deverão estar na região Leste dos EUA (Ohio).

Para obter mais informações, consulte [Creating a Bucket](#) (Criar um bucket) no Manual do usuário do Amazon Simple Storage Service.

Note

Embora CodeBuild também ofereça suporte a entradas de compilação armazenadas nos CodeCommit repositórios,, e do Bitbucket, este tutorial não mostra como usá-las. GitHub Para obter mais informações, consulte [Planejar uma compilação](#).

Etapa 4: fazer upload do código-fonte e do arquivo buildspec

(Etapa anterior: [Etapa 3: Criar dois buckets do S3](#))

Nesta etapa, você adiciona o código-fonte e o arquivo de build spec ao bucket de entrada.

Usando o utilitário zip do sistema operacional, crie um arquivo chamado `MessageUtil.zip` que inclua `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` e `buildspec.yml`.

A estrutura de diretório de arquivo `MessageUtil.zip` deve se parecer como esta.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```


 Important

Não inclua o diretório (*root directory name*), somente os diretórios e os arquivos no diretório (*root directory name*).

Faça upload do arquivo `MessageUtil.zip` ao bucket de entrada denominado `codebuild-region-ID-account-ID-input-bucket`.

 Important

Para repositórios CodeCommit, GitHub, e Bitbucket, por convenção, você deve armazenar um arquivo de especificação de construção nomeado `buildspec.yml` na raiz (nível superior) de cada repositório ou incluir a declaração de especificação de construção como parte da definição do projeto de construção. Não crie um arquivo ZIP que contenha o código-fonte do repositório e o arquivo build spec.

Para a entrada de compilação armazenada apenas em buckets do S3, crie um arquivo ZIP que contenha o código-fonte e, por convenção, um arquivo de especificação da compilação chamado `buildspec.yml` na raiz (nível superior) ou inclua a declaração da especificação da compilação como parte da definição do projeto de compilação.

Se quiser usar um nome diferente para o arquivo de especificação da compilação, ou fazer referência a uma especificação da compilação em um local diferente da raiz, você poderá definir uma substituição da especificação de compilação como parte da definição do projeto de compilação. Para obter mais informações, consulte [Nome do arquivo buildspec e local de armazenamento](#).

Etapa 5: criar o projeto de compilação

(Etapa anterior: [Etapa 4: fazer upload do código-fonte e do arquivo buildspec](#))

Nesta etapa, você cria um projeto de compilação que AWS CodeBuild usa para executar a compilação. Um projeto de compilação inclui informações sobre como executar uma compilação, incluindo onde obter o código-fonte, qual ambiente de compilação usar, quais comandos de compilação executar e onde armazenar a saída da compilação. Um ambiente de compilação representa uma combinação de sistema operacional, tempo de execução da linguagem de programação e ferramentas CodeBuild usadas para executar uma compilação. O ambiente de

compilação é expresso como uma imagem do Docker. Para obter mais informações, consulte [Visão geral do Docker](#) no site Docker Docs.

Para esse ambiente de compilação, você instrui CodeBuild a usar uma imagem do Docker que contém uma versão do Java Development Kit (JDK) e do Apache Maven.

Como criar o projeto de compilação

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Use o seletor de AWS região para escolher uma AWS região em que CodeBuild haja suporte. Para obter mais informações, consulte [endpoints e cotas do AWS CodeBuild](#) na Referência geral da Amazon Web Services.
3. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
4. Na página Create build project (Criar projeto de compilação), em Project Configuration (Configuração do projeto), em Project name (Nome do projeto), insira um nome para esse projeto de compilação (neste exemplo, codebuild-demo-project). Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Se você escolher outro nome, certifique-se de usá-lo durante todo este tutorial.

Note

Na página Create build project (Criar projeto de compilação), você pode ver uma mensagem de erro semelhante à seguinte: You are not authorized to perform this operation (Você não tem autorização para realizar esta operação). Provavelmente, isso ocorre porque você fez login no AWS Management Console como um usuário que não tem permissões para criar um projeto de compilação. Para corrigir isso, saia do e AWS Management Console, em seguida, entre novamente com credenciais pertencentes a uma das seguintes entidades do IAM:

- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
- Um usuário em sua AWS conta com as políticas `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, e `IAMFullAccess` gerenciadas anexadas a esse usuário ou a um grupo do IAM ao

qual o usuário pertence. Se você não tiver um usuário ou grupo em sua AWS conta com essas permissões e não puder adicionar essas permissões ao seu usuário ou grupo, entre em contato com o administrador AWS da conta para obter ajuda. Para obter mais informações, consulte [AWS políticas gerenciadas \(predefinidas\) para AWS CodeBuild](#).

Ambas as opções incluem permissões de administrador que permitem criar um projeto de compilação para que você possa concluir este tutorial. Recomendamos que você sempre use as permissões mínimas necessárias para realizar sua tarefa. Para obter mais informações, consulte [AWS CodeBuild referência de permissões](#).

5. Em Origem, para Provedor de origem, selecione Amazon S3.
6. Para Bucket, escolha codebuild- **region-ID** - **account-ID** -input-bucket.
7. Para S3 object key (Chave do objeto S3), insira **MessageUtil.zip**.
8. Em Environment (Ambiente), em Environment image (Imagem do ambiente), deixe a opção Managed image (Imagem gerenciada) selecionada.
9. Em Sistema operacional, escolha Amazon Linux.
10. Em Runtime(s), selecione Padrão.
11. Para Imagem, escolha aws/codebuild/amazonlinux-x86_64-standard:corretto11.
12. Em Service role (Função de serviço), deixe New service role (Nova função de serviço) selecionado e não altere Role name (Nome da função).
13. Em Buildspec, deixe Use a buildspec file (Usar um arquivo buildspec) selecionado.
14. Em Artefatos, em Tipo, escolha Amazon S3.
15. Em Nome do bucket, escolha codebuild- **region-ID** - **account-ID** -output-bucket.
16. Deixe Name (Nome) e Path (Caminho) em branco.
17. Selecione Create build project (Criar projeto de compilação).

Etapa 6: executar a compilação

(Etapa anterior: [Etapa 5: criar o projeto de compilação](#))

Nesta etapa, você instrui AWS CodeBuild a executar a compilação com as configurações do projeto de compilação.

Para executar a compilação

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Na lista de projetos de compilação, escolha `codebuild-demo-project`, em seguida, escolha Iniciar compilação. A compilação começa imediatamente.

Etapa 7: visualizar informações resumidas da compilação

(Etapa anterior: [Etapa 6: executar a compilação](#))

Nesta etapa, você exibe informações resumidas sobre o status de seu build.

Como visualizar informações resumidas da compilação

1. Se a `<build-ID>` página `codebuild-demo-project`: não for exibida, na barra de navegação, escolha Histórico de compilação. Em seguida, na lista de projetos de compilação, para Projeto, escolha o link Construir execução para `codebuild-demo-project`. Deve haver somente um link correspondente. (Se você tiver concluído este tutorial anteriormente, escolha o link com o valor mais recente na coluna Completed (Concluído).)
2. Na página Status da compilação, em Detalhes da fase, as seguintes fases de compilação devem ser exibidas, com Com êxito na coluna Status:
 - SUBMITTED
 - QUEUED
 - PROVISIONING
 - DOWNLOAD_SOURCE
 - INSTALL
 - PRE_BUILD
 - BUILD
 - POST_BUILD
 - UPLOAD_ARTIFACTS
 - FINALIZING
 - COMPLETED

Em Build Status (Status da compilação), Succeeded (Bem-sucedido) deve ser exibido.

Se você vir In Progress (Em Andamento), escolha o botão Atualizar.

3. Ao lado de cada fase de compilação, o valor Duration (Duração) indica quanto tempo a fase de compilação demorou. O valor End time (Hora de término) indica quando a fase de build foi concluída.

Etapa 8: visualizar informações detalhadas da compilação

(Etapa anterior: [Etapa 7: visualizar informações resumidas da compilação](#))

Nesta etapa, você visualiza informações detalhadas sobre sua compilação no CloudWatch Logs.

Note

Para proteger informações confidenciais, o seguinte está oculto nos CodeBuild registros:

- AWS chave de acesso IDs. Para obter mais informações, consulte [Managing Access Keys for IAM Users](#) no Guia do usuário do AWS Identity and Access Management .
- Strings especificadas usando o repositório de parâmetros. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.
- Cadeias de caracteres especificadas usando AWS Secrets Manager. Para obter mais informações, consulte [Gerenciamento de chaves](#).

Como visualizar informações detalhadas da compilação

1. Com a página de detalhes de build ainda em exibição na etapa anterior, as últimas 10,000 linhas do log de build são exibidas em Build logs. Para ver todo o registro de compilação em CloudWatch Registros, escolha o link Exibir registro inteiro.
2. No stream de CloudWatch registros de registros, você pode navegar pelos eventos de registro. Por padrão, somente o último conjunto de eventos de log é exibido. Para ver eventos de log anteriores, role até o início da lista.
3. Neste tutorial, a maioria dos eventos de log contém informações detalhadas sobre o download e a instalação pelo CodeBuild dos arquivos de dependência da compilação em seu ambiente

de compilação, informações essas que não são tão importantes pra você. Você pode usar a caixa Filter events para reduzir as informações exibidas. Por exemplo, se você inserir "[INFO]" em Filter events (Filtrar eventos), apenas os eventos que contêm [INFO] serão exibidos. Para obter mais informações, consulte [Sintaxe de filtros e padrões](#) no Guia do CloudWatch usuário da Amazon.

Etapa 9: obter o artefato de saída da compilação

(Etapa anterior: [Etapa 8: visualizar informações detalhadas da compilação](#))

Nesta etapa, você obtém o messageUtil-1.0.jar arquivo CodeBuild criado e carregado no bucket de saída.

Você pode usar o CodeBuild console ou o console do Amazon S3 para concluir essa etapa.

Para obter o artefato de saída da compilação (AWS CodeBuild console)

1. Com o CodeBuild console ainda aberto e a página de detalhes da construção ainda exibida na etapa anterior, escolha a guia Detalhes da construção e role para baixo até a seção Artefatos.

Note

Se a página de detalhes da compilação não foi exibida, na barra de navegação, selecione Histórico de compilação e escolha o link Execução da compilação.

2. O link para a pasta do Amazon S3 está abaixo do Local de upload de artefatos. Esse link abre a pasta no Amazon S3 em que você encontra o arquivo de artefato de saída da compilação messageUtil-1.0.jar.

Como obter o artefato de saída da compilação (console do Amazon S3)

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Abra o codebuild-*region-ID-account-ID*-output-bucket.
3. Abra a pasta codebuild-demo-project.
4. Abra a pasta target, onde você encontra o arquivo de artefato de saída da compilação messageUtil-1.0.jar.

Etapa 10: Excluir os buckets do S3

(Etapa anterior: [Etapa 9: obter o artefato de saída da compilação](#))

Para evitar cobranças contínuas em sua AWS conta, você pode excluir os intervalos de entrada e saída usados neste tutorial. Para obter instruções, consulte [Deleting or Emptying a Bucket](#) no Guia do usuário do Amazon Simple Storage Service.

Se você estiver utilizando o usuário do IAM ou um usuário administrador do IAM para excluir esses buckets, o usuário deverá ter mais permissões de acesso. Adicione a seguinte declaração entre os marcadores (**### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENTS HERE ###**) a uma política de acesso existente para o usuário.

As elipses (...) nesta declaração são usadas para fins de brevidade. Não remova as declarações na política de acesso existente. Não insira essas reticências na política.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Encapsulamento

Neste tutorial, você AWS CodeBuild costumava criar um conjunto de arquivos de classe Java em um arquivo JAR. Em seguida, você exibiu os resultados do build.

Agora você pode tentar usar CodeBuild em seus próprios cenários. Siga as instruções em [Planejar uma compilação](#). Se não se sente pronto ainda, tente montar algumas das amostras. Para obter mais informações, consulte [Use amostras baseadas em casos para CodeBuild](#).

Começando a AWS CodeBuild usar o AWS CLI

Neste tutorial, você usa AWS CodeBuild para criar uma coleção de exemplos de arquivos de entrada de código-fonte (chamados de artefatos de entrada de compilação ou entrada de compilação) em uma versão implantável do código-fonte (chamada artefato de saída de compilação ou saída de construção). Especificamente, você instrui CodeBuild a usar o Apache Maven, uma ferramenta de construção comum, para criar um conjunto de arquivos de classe Java em um arquivo Java Archive (JAR). Você não precisa estar familiarizado com o Apache Maven ou com o Java para utilizar este tutorial.

Você pode trabalhar com CodeBuild ele por meio do CodeBuild console AWS CodePipeline, AWS CLI, do ou do AWS SDKs. Este tutorial demonstra como usar CodeBuild com o. AWS CLI Para obter informações sobre o uso CodePipeline, consulte [Use CodeBuild com CodePipeline](#).

Important

As etapas deste tutorial exigem que você crie recursos (por exemplo, um bucket do S3) que podem resultar em cobranças na sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados ao Amazon S3 e CloudWatch aos AWS KMS registros. Para obter mais informações, consulte [CodeBuild preços, preços](#) do [Amazon S3, preços](#) e [AWS Key Management Service preços](#) da [Amazon CloudWatch](#).

Tópicos

- [Etapa 1: Criar o código-fonte](#)
- [Etapa 2: Criar o arquivo buildspec](#)
- [Etapa 3: Criar dois buckets do S3](#)
- [Etapa 4: fazer upload do código-fonte e do arquivo buildspec](#)
- [Etapa 5: criar o projeto de compilação](#)
- [Etapa 6: executar a compilação](#)
- [Etapa 7: visualizar informações resumidas da compilação](#)
- [Etapa 8: visualizar informações detalhadas da compilação](#)
- [Etapa 9: obter o artefato de saída da compilação](#)
- [Etapa 10: Excluir os buckets do S3](#)
- [Encapsulamento](#)

Etapa 1: Criar o código-fonte

(Parte de: [Começando a AWS CodeBuild usar o AWS CLI](#))

Nesta etapa, você cria o código-fonte que CodeBuild deseja criar no bucket de saída. O código-fonte consiste em dois arquivos classe Java e um arquivo Apache Maven Project Object Model (POM).

1. Em um diretório vazio em seu computador ou instância local, crie essa estrutura de diretório.

```
(root directory name)
|-- src
|   |-- main
|       |-- java
|       |-- test
|           |-- java
```

2. Use um editor de texto de sua preferência, crie esse arquivo, nomeie-o como `MessageUtil.java` e salve-o no diretório `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }

    public String salutationMessage() {
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

Esse arquivo classe cria como a saída o string de caracteres para ele transferido. O construtor `MessageUtil` define o string de caracteres. O método `printMessage` cria a saída. O método `salutationMessage` gera a saída `Hi!` seguida da string de caracteres.

3. Crie esse arquivo, nomeie-o `TestMessageUtil.java` e o salve no diretório `/src/test/java`.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message,messageUtil.salutationMessage());
    }
}
```

Esse arquivo classe define a variável `message` na classe `MessageUtil` para Robert. O arquivo avalia, então, se a variável `message` foi definida com sucesso, verificando se as strings Robert e Hi!Robert aparecem na saída.

4. Crie esse arquivo, nomeie-o `pom.xml` e salve-o no diretório raiz (de nível superior).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>messageUtil</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
    <name>Message Utility Java Sample App</name>
    <dependencies>
```

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.11</version>
  <scope>test</scope>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.0</version>
    </plugin>
  </plugins>
</build>
</project>
```

O Apache Maven usa as instruções nesse arquivo para converter os arquivos `MessageUtil.java` e `TestMessageUtil.java` em um arquivo denominado `messageUtil-1.0.jar` e, em seguida, executar os testes especificados.

Neste momento, a estrutura de diretório deve se parecer como essa.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Etapa 2: Criar o arquivo buildspec

(Etapa anterior: [Etapa 1: Criar o código-fonte](#))

Nesta etapa, você cria um arquivo de especificação de build (build spec). Um buildspec é uma coleção de comandos de compilação e configurações relacionadas, no formato YAML, CodeBuild usados para executar uma compilação. Sem uma especificação de compilação, CodeBuild não é

possível converter com êxito sua entrada de compilação em saída de compilação ou localizar o artefato de saída de compilação no ambiente de compilação para fazer upload em seu bucket de saída.

Crie esse arquivo, nomeie-o `buildspec.yml` e salve-o no diretório raiz (de nível superior).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Como a declaração de build spec deve corresponder ao YAML, o espaçamento na declaração de build spec é importante. Se o número de espaços em sua declaração de build spec não corresponder a essa, o build pode falhar imediatamente. Você pode usar um validador YAML para testar se a declaração de build spec é válida para o YAML.

Note

Em vez de incluir um arquivo de build spec em seu código-fonte, você pode declarar comandos de build separadamente ao criar um projeto de build. Isso é útil no caso de você compilar seu código-fonte com diferentes comandos de build sem atualizar o repositório de código fonte a cada vez. Para obter mais informações, consulte [Sintaxe de buildspec](#).

Nesta declaração de build spec:

- `version` representa a versão do build spec padrão em uso. Essa declaração de build spec usa a última versão, `0.2`.
- `phases` representa as fases do build durante as quais você pode instruir o CodeBuild para executar comandos. Essas fases de build são aqui listadas como `install`, `pre_build`, `build` e `post_build`. Não é possível alterar os nomes das fases de build, nem criar nomes adicionais.

Neste exemplo, durante a `build` fase, CodeBuild executa o `mvn install` comando. Esse comando instrui o Apache Maven a compilar, testar e empacotar os arquivos classe Java em um artefato de saída do build. Para completar, alguns comandos `echo` são incluídos em cada fase de build, neste exemplo. Quando você visualizar informações de compilação detalhadas posteriormente neste tutorial, a saída desses comandos `echo` poderá ajudar a compreender melhor como o CodeBuild executa comandos e em que ordem. (Embora todas as fases da compilação estejam incluídas neste exemplo, não é necessário incluir uma fase da compilação se você não pretende executar comandos durante essa fase.) Para cada fase de compilação, CodeBuild executa cada comando especificado, um por vez, na ordem listada, do início ao fim.

- `artifacts` representa o conjunto de artefatos de saída de compilação que são CodeBuild carregados no bucket de saída. `files` representa os arquivos a serem incluídos na saída da compilação. CodeBuild carrega o único `messageUtil-1.0.jar` arquivo encontrado no diretório `target` relativo no ambiente de compilação. O nome de arquivo `messageUtil-1.0.jar` e o nome de diretório `target` são baseados na maneira como o Apache Maven cria e armazena artefatos de saída do build somente para este exemplo. Em seus próprios builds, esses nomes de arquivos e diretórios são diferentes.

Para obter mais informações, consulte o [Referência de buildspec](#).

Neste momento, a estrutura de diretório deve se parecer como essa.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |    |-- java
    |    |-- MessageUtil.java
    |-- test
    |    |-- java
```

```
`-- TestMessageUtil.java
```

Etapa 3: Criar dois buckets do S3

(Etapa anterior: [Etapa 2: Criar o arquivo buildspec](#))

Embora seja possível usar um único bucket para este tutorial, o uso de dois buckets facilita a visualização da origem da entrada da compilação e do destino da saída da compilação.

- Um desses buckets (o bucket de entrada) armazena a entrada da compilação. Neste tutorial, o nome desse bucket de entrada é `codebuild-region-ID-account-ID-input-bucket`, onde *region-ID* é a AWS região do bucket e *account-ID* é o ID da sua AWS conta.
- O outro bucket (o bucket de saída) armazena a saída da compilação. Neste tutorial, o nome deste bucket de saída é `codebuild-region-ID-account-ID-output-bucket`.

Se você escolheu nomes diferentes para esses buckets, certifique-se de usá-los ao longo deste tutorial.

Esses dois buckets devem estar na mesma AWS região das suas compilações. Por exemplo, se você CodeBuild instruir a execução de uma compilação na região Leste dos EUA (Ohio), esses buckets também deverão estar na região Leste dos EUA (Ohio).

Para obter mais informações, consulte [Creating a Bucket](#) (Criar um bucket) no Manual do usuário do Amazon Simple Storage Service.

Note

Embora CodeBuild também ofereça suporte a entradas de compilação armazenadas nos CodeCommit repositórios, e do Bitbucket, este tutorial não mostra como usá-las. GitHub Para obter mais informações, consulte [Planejar uma compilação](#).

Etapa 4: fazer upload do código-fonte e do arquivo buildspec

(Etapa anterior: [Etapa 3: Criar dois buckets do S3](#))

Nesta etapa, você adiciona o código-fonte e o arquivo de build spec ao bucket de entrada.

Usando o utilitário zip do sistema operacional, crie um arquivo chamado `MessageUtil.zip` que inclua `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` e `buildspec.yml`.

A estrutura de diretório de arquivo MessageUtil.zip deve se parecer como esta.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

 Important

Não inclua o diretório (*root directory name*), somente os diretórios e os arquivos no diretório (*root directory name*).

Faça upload do arquivo MessageUtil.zip ao bucket de entrada denominado codebuild-*region-ID-account-ID*-input-bucket.

 Important

Para repositórios CodeCommit GitHub,, e Bitbucket, por convenção, você deve armazenar um arquivo de especificação de construção nomeado buildspec.yml na raiz (nível superior) de cada repositório ou incluir a declaração de especificação de construção como parte da definição do projeto de construção. Não crie um arquivo ZIP que contenha o código-fonte do repositório e o arquivo build spec.

Para a entrada de compilação armazenada apenas em buckets do S3, crie um arquivo ZIP que contenha o código-fonte e, por convenção, um arquivo de especificação da compilação chamado buildspec.yml na raiz (nível superior) ou inclua a declaração da especificação da compilação como parte da definição do projeto de compilação.

Se quiser usar um nome diferente para o arquivo de especificação da compilação, ou fazer referência a uma especificação da compilação em um local diferente da raiz, você poderá definir uma substituição da especificação de compilação como parte da definição do projeto de compilação. Para obter mais informações, consulte [Nome do arquivo buildspec e local de armazenamento](#).

Etapa 5: criar o projeto de compilação

(Etapa anterior: [Etapa 4: fazer upload do código-fonte e do arquivo buildspec](#))

Nesta etapa, você cria um projeto de compilação que AWS CodeBuild usa para executar a compilação. Um projeto de compilação inclui informações sobre como executar uma compilação, incluindo onde obter o código-fonte, qual ambiente de compilação usar, quais comandos de compilação executar e onde armazenar a saída da compilação. Um ambiente de compilação representa uma combinação de sistema operacional, tempo de execução da linguagem de programação e ferramentas CodeBuild usadas para executar uma compilação. O ambiente de compilação é expresso como uma imagem do Docker. Para obter mais informações, consulte [Visão geral do Docker](#) no site Docker Docs.

Para esse ambiente de compilação, você instrui CodeBuild a usar uma imagem do Docker que contém uma versão do Java Development Kit (JDK) e do Apache Maven.

Como criar o projeto de compilação

1. Use o AWS CLI para executar o create-project comando:

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

Os dados formatados JSON aparecem na saída. Copie os dados para um arquivo nomeado `create-project.json` em um local no computador local ou na instância em que o AWS CLI está instalado. Se você optar por usar um nome de arquivo diferente, certifique-se de usá-lo durante todo este tutorial.

Altere os dados copiados para acompanhar o formato e, em seguida, salve os resultados:

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
```



```
"image": "aws/codebuild/standard:5.0",
"computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "serviceIAMRole"
}
```

serviceIAMRole Substitua pelo Amazon Resource Name (ARN) de uma função de CodeBuild serviço (por exemplo, `arn:aws:iam::account-ID:role/role-name`). Para criar uma, consulte [CodeBuild Permitir interagir com outros AWS serviços](#).

Nesses dados:

- `name` representa um identificador necessário para o projeto de build (neste exemplo, `codebuild-demo-project`). Os nomes de projeto de build devem ser únicos em todos os projetos de build em sua conta.
- Para `source`, `type` é um valor necessário que representa o tipo de repositório de código-fonte (neste exemplo, `S3` para um bucket do Amazon S3).
- Para `source`, `location` representa o caminho para o código-fonte (neste exemplo, o nome de bucket de entrada seguido pelo nome do arquivo ZIP).
- Para `artifacts`, `type` é um valor obrigatório que representa o tipo de repositório de artefato de saída da compilação (neste exemplo, `S3` para um bucket do Amazon S3).
- Para `artifacts`, `location` representa o nome do bucket de saída que você criou ou identificou anteriormente (neste exemplo, `codebuild-region-ID-account-ID-output-bucket`).
- Para `environment`, `type` é um valor obrigatório que representa o tipo de ambiente de compilação (neste exemplo, `LINUX_CONTAINER`).
- Para `environment`, `image` é um valor obrigatório que representa a combinação de nome de imagem e tag do Docker que esse projeto de construção usa, conforme especificado pelo tipo de repositório de imagens do Docker (neste exemplo, `aws/codebuild/standard:5.0` para uma imagem do Docker no repositório de imagens do CodeBuild Docker). `aws/codebuild/standard` é o nome da imagem do Docker. `5.0` é a tag da imagem do Docker.

Para encontrar mais imagens Docker que você possa utilizar em seus cenários, consulte [Referência de ambiente de compilação](#).

- Para `environment`, `computeType` é um valor obrigatório que representa o CodeBuild uso dos recursos de computação (neste exemplo, `BUILD_GENERAL1_SMALL`).

 Note

Outros valores disponíveis nos dados formatados JSON originais, tais como `description`, `buildspec`, `auth` (incluindo `type` e `resource`), `path`, `namespaceType`, `name` (para `artifacts`), `packaging`, `environmentVariables` (incluindo `name` e `value`), `timeoutInMinutes`, `encryptionKey` e `tags` (incluindo `key` e `value`) são opcionais. Eles não são usados neste tutorial, portanto, não são mostrados aqui. Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).

2. Alterne para o diretório que contenha o arquivo que você acabou de salvar e execute o comando `create-project` novamente.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Se bem-sucedidos, dados semelhantes a estes aparecem na saída.

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
```

```
{
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
"arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
}
```

- `project` representa informações sobre esse projeto de build.
- `tags` representa as tags que foram declaradas.
- `packaging` representa como o artefato de saída de compilação é armazenado no bucket de saída. `NONE` significa que uma pasta foi criada no bucket de saída. O artefato de saída de compilação foi armazenado na pasta.
- `lastModified` representa quando as informações sobre o projeto de build foram modificadas pela última vez, em formato de data e hora Unix.
- `timeoutInMinutes` representa o número de minutos após os quais CodeBuild interrompe a construção se a construção não tiver sido concluída. (O padrão é 60 minutos.)
- `created` representa quando o projeto de build foi criado, em formato de data e hora Unix.
- `environmentVariables` representa todas as variáveis de ambiente que foram declaradas e estão disponíveis CodeBuild para uso durante a construção.
- `encryptionKey` representa o ARN da chave gerenciada pelo cliente CodeBuild usada para criptografar o artefato de saída da compilação.
- `arn` representa o ARN do projeto de build.

Note

Depois de executar o `create-project` comando, uma mensagem de erro semelhante à seguinte pode ser gerada: Usuário: não ***user-ARN*** está autorizado a executar: `codebuild:CreateProject` Provavelmente, isso ocorre porque você configurou o AWS CLI com as credenciais de um usuário que não tem permissões suficientes CodeBuild para usar na criação de projetos de compilação. Para corrigir isso, configure a AWS CLI com credenciais pertencentes a uma das seguintes entidades do IAM:

- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
- Um usuário em sua AWS conta com as políticas `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, e `IAMFullAccess`

gerenciadas anexadas a esse usuário ou a um grupo do IAM ao qual o usuário pertence. Se você não tiver um usuário ou grupo em sua AWS conta com essas permissões e não puder adicionar essas permissões ao seu usuário ou grupo, entre em contato com o administrador AWS da conta para obter ajuda. Para obter mais informações, consulte [AWS políticas gerenciadas \(predefinidas\) para AWS CodeBuild](#).

Etapa 6: executar a compilação

(Etapa anterior: [Etapa 5: criar o projeto de compilação](#))

Nesta etapa, você instrui AWS CodeBuild a executar a compilação com as configurações do projeto de compilação.

Para executar a compilação

1. Use o AWS CLI para executar o start-build comando:

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

project-name Substitua pelo nome do seu projeto de construção da etapa anterior (por exemplo, codebuild-demo-project).

2. Se bem-sucedido, dados semelhantes aos seguintes serão exibidos na saída:

```
{
  "build": {
    "buildComplete": false,
    "initiator": "user-name",
    "artifacts": {
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/
message-util.zip"
    },
    "projectName": "codebuild-demo-project",
    "timeoutInMinutes": 60,
    "buildStatus": "IN_PROGRESS",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    }
  }
}
```

```
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "SUBMITTED",
    "startTime": 1472848787.882,
    "id": "codebuild-demo-project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE"
  }
}
```

- `build` representa informações sobre este build.
- `buildComplete` representa se a compilação foi concluída (`true`). Caso contrário, `false`.
- `initiator` representa a entidade que iniciou o build.
- `artifacts` representa informações sobre a saída de build, incluindo seu local.
- `projectName` representa o nome do projeto de build.
- `buildStatus` representa o status atual do build, quando o comando `start-build` foi executado.
- `currentPhase` representa a fase atual do build quando o comando `start-build` foi executado.
- `startTime` representa o momento em que o processo de build foi iniciado, em formato de data e hora do Unix.
- `id` representa a ID do build.
- `arn` representa o ARN do build.

Anote o valor do `id`. Você precisa dele na próxima etapa.

Etapa 7: visualizar informações resumidas da compilação

(Etapa anterior: [Etapa 6: executar a compilação](#))

Nesta etapa, você exibe informações resumidas sobre o status de seu build.

Como visualizar informações resumidas da compilação

- Use o AWS CLI para executar o batch-get-builds comando.

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

id Substitua pelo id valor que apareceu na saída da etapa anterior.

Se bem-sucedidos, dados semelhantes a estes aparecem na saída.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
          "startTime": 1472848787.882
        },
        ... The full list of build phases has been omitted for brevity ...
        {
          "phaseType": "COMPLETED",
          "startTime": 1472848878.079
        }
      ],
      "logs": {
        "groupName": "/aws/codebuild/codebuild-demo-project",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
        "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
      },
      "artifacts": {
        "md5sum": "MD5-hash",
        "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/message-util.zip",
        "sha256sum": "SHA-256-hash"
      },
      "projectName": "codebuild-demo-project",
    }
  ]
}
```

```

    "timeoutInMinutes": 60,
    "initiator": "user-name",
    "buildStatus": "SUCCEEDED",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "COMPLETED",
    "startTime": 1472848787.882,
    "endTime": 1472848878.079,
    "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
  }
]
}

```

- `buildsNotFound` representa a construção IDs de qualquer compilação em que as informações não estejam disponíveis. Neste exemplo, deve estar vazia.
- `builds` representa informações sobre cada build em que haja informações disponíveis. Neste exemplo, informações sobre somente um build aparece na saída.
 - `phases` representa o conjunto de fases de build que o CodeBuild executa durante o processo de build. Informações sobre cada fase de build é listada separadamente como `startTime`, `endTime`, e `durationInSeconds` (quando a fase de build iniciou e terminou, expressa em formato de data e hora do Unix, e quanto tempo durou, em segundos) e `phaseType`, tais como (SUBMITTED, PROVISIONING, DOWNLOAD_SOURCE, INSTALL, PRE_BUILD, BUILD, POST_BUILD, UPLOAD_ARTIFACTS, FINALIZING ou COMPLETED) e `phaseStatus` (tais como SUCCEEDED, FAILED, FAULT, TIMED_OUT, IN_PROGRESS ou STOPPED). Na primeira vez que você executar o comando `batch-get-builds`, pode não haver muitas fases (ou nenhuma). Depois de execuções subsequentes do comando `batch-get-builds` com a mesma ID de build, mais fases de build deverão aparecer na saída.
 - `logs` representa informações no Amazon CloudWatch Logs sobre os registros da compilação.

- `md5sum` MD5 e `sha256sum` representam os hashes SHA-256 do artefato de saída da compilação. Esses aparecem na saída somente se o `packaging` valor do projeto de compilação estiver definido como ZIP. (Você não definiu esse valor neste tutorial.) Você pode usar esses hashes em conjunto com uma ferramenta de checksum para confirmar a integridade e a autenticidade dos arquivos.

 Note

Também é possível usar o console do Amazon S3 para exibir esses hashes. Marque a caixa próxima ao artefato de saída da compilação e escolha **Actions** (Ações) e **Properties** (Propriedades). No painel **Propriedades**, expanda **Metadados** e visualize os valores de `-content-md5` e `x-amz-meta-codebuild-content-sha256`. `x-amz-meta-codebuild` (No console do Amazon S3, o ETag valor do artefato de saída de compilação não deve ser interpretado como o hash MD5 ou SHA-256.) Se você usar o AWS SDKs para obter esses hashes, os valores serão nomeados `codebuild-content-md5` e `codebuild-content-sha256`.

- `endTime` representa o momento em que o processo de build foi concluído, em formato de data e hora do Unix.

 Note

Os metadados do Amazon S3 têm um CodeBuild cabeçalho chamado `x-amz-meta-codebuild-buildarn` que contém o `buildArn` da CodeBuild compilação que publica artefatos no Amazon S3. O `buildArn` é adicionado para permitir o rastreamento da fonte para notificações e para referenciar de qual compilação o artefato é gerado.

Etapa 8: visualizar informações detalhadas da compilação

(Etapa anterior: [Etapa 7: visualizar informações resumidas da compilação](#))

Nesta etapa, você visualiza informações detalhadas sobre sua compilação no CloudWatch Logs.

 Note

Para proteger informações confidenciais, o seguinte está oculto nos CodeBuild registros:

- AWS chave de acesso IDs. Para obter mais informações, consulte [Managing Access Keys for IAM Users](#) no Guia do usuário do AWS Identity and Access Management .
- Strings especificadas usando o repositório de parâmetros. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.
- Cadeias de caracteres especificadas usando AWS Secrets Manager. Para obter mais informações, consulte [Gerenciamento de chaves](#).

Como visualizar informações detalhadas da compilação

1. Use seu navegador para acessar o local deepLink que apareceu na saída na etapa anterior (por exemplo, `https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).
2. No stream de CloudWatch registros de registros, você pode navegar pelos eventos de registro. Por padrão, somente o último conjunto de eventos de log é exibido. Para ver eventos de log anteriores, role até o início da lista.
3. Neste tutorial, a maioria dos eventos de log contém informações detalhadas sobre o download e a instalação pelo CodeBuild dos arquivos de dependência da compilação em seu ambiente de compilação, informações essas que não são tão importantes pra você. Você pode usar a caixa Filter events para reduzir as informações exibidas. Por exemplo, se você inserir "[INFO]" em Filter events (Filtrar eventos), apenas os eventos que contêm [INFO] serão exibidos. Para obter mais informações, consulte [Sintaxe de filtros e padrões](#) no Guia do CloudWatch usuário da Amazon.

Essas partes de um stream de CloudWatch registros de registros pertencem a este tutorial.

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
```

```
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
[Container] 2016/04/15 17:49:44 [INFO]
[Container] 2016/04/15 17:49:44 [INFO]
-----
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
-----
...
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 T E S T S
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 Running TestMessageUtil
[Container] 2016/04/15 17:49:55 Inside testSalutationMessage()
[Container] 2016/04/15 17:49:55 Hi!Robert
[Container] 2016/04/15 17:49:55 Inside testPrintMessage()
[Container] 2016/04/15 17:49:55 Robert
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.018 sec
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Results :
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
...
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] BUILD SUCCESS
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] Total time: 11.845 s
[Container] 2016/04/15 17:49:56 [INFO] Finished at: 2016-04-15T17:49:56+00:00
[Container] 2016/04/15 17:49:56 [INFO] Final Memory: 18M/216M
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 Phase complete: BUILD Success: true
[Container] 2016/04/15 17:49:56 Entering phase POST_BUILD
[Container] 2016/04/15 17:49:56 Running command echo Entering post_build phase...
[Container] 2016/04/15 17:49:56 Entering post_build phase...
[Container] 2016/04/15 17:49:56 Phase complete: POST_BUILD Success: true
[Container] 2016/04/15 17:49:57 Preparing to copy artifacts
[Container] 2016/04/15 17:49:57 Assembling file list
[Container] 2016/04/15 17:49:57 Expanding target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Found target/messageUtil-1.0.jar
```

```
[Container] 2016/04/15 17:49:57 Creating zip artifact
```

Neste exemplo, concluiu CodeBuild com êxito as fases de pré-construção, construção e pós-construção. Ele executou os testes da unidade e montou com sucesso o arquivo `messageUtil-1.0.jar`.

Etapa 9: obter o artefato de saída da compilação

(Etapa anterior: [Etapa 8: visualizar informações detalhadas da compilação](#))

Nesta etapa, você obtém o `messageUtil-1.0.jar` arquivo CodeBuild criado e carregado no bucket de saída.

Você pode usar o CodeBuild console ou o console do Amazon S3 para concluir essa etapa.

Para obter o artefato de saída da compilação (AWS CodeBuild console)

1. Com o CodeBuild console ainda aberto e a página de detalhes da construção ainda exibida na etapa anterior, escolha a guia Detalhes da construção e role para baixo até a seção Artefatos.

Note

Se a página de detalhes da compilação não foi exibida, na barra de navegação, selecione Histórico de compilação e escolha o link Execução da compilação.

2. O link para a pasta do Amazon S3 está abaixo do Local de upload de artefatos. Esse link abre a pasta no Amazon S3 em que você encontra o arquivo de artefato de saída da compilação `messageUtil-1.0.jar`.

Como obter o artefato de saída da compilação (console do Amazon S3)

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Abra o `codebuild-region-ID-account-ID-output-bucket`.
3. Abra a pasta `codebuild-demo-project`.
4. Abra a pasta `target`, onde você encontra o arquivo de artefato de saída da compilação `messageUtil-1.0.jar`.

Etapa 10: Excluir os buckets do S3

(Etapa anterior: [Etapa 9: obter o artefato de saída da compilação](#))

Para evitar cobranças contínuas em sua AWS conta, você pode excluir os intervalos de entrada e saída usados neste tutorial. Para obter instruções, consulte [Deleting or Emptying a Bucket](#) no Guia do usuário do Amazon Simple Storage Service.

Se você estiver utilizando o usuário do IAM ou um usuário administrador do IAM para excluir esses buckets, o usuário deverá ter mais permissões de acesso. Adicione a seguinte declaração entre os marcadores (**### BEGIN ADDING STATEMENT HERE ###**e**### END ADDING STATEMENTS HERE ###**) a uma política de acesso existente para o usuário.

As elipses (...) nesta declaração são usadas para fins de brevidade. Não remova as declarações na política de acesso existente. Não insira essas reticências na política.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Encapsulamento

Neste tutorial, você AWS CodeBuild costumava criar um conjunto de arquivos de classe Java em um arquivo JAR. Em seguida, você exibiu os resultados do build.

Agora você pode tentar usar CodeBuild em seus próprios cenários. Siga as instruções em [Planejar uma compilação](#). Se não se sente pronto ainda, tente montar algumas das amostras. Para obter mais informações, consulte [Use amostras baseadas em casos para CodeBuild](#).

Use amostras baseadas em casos para CodeBuild

Você pode usar essas amostras baseadas em casos de uso para fazer experiências com: AWS CodeBuild

[Exemplos entre serviços](#)

Uma lista de amostras de serviços cruzados para experimentar AWS CodeBuild.

[Exemplo de emblemas de compilação](#)

Mostra como configurar CodeBuild com emblemas de construção.

[Exemplo de relatório de teste](#)

Usa o AWS CLI para criar, executar e visualizar os resultados de um relatório de teste.

[Amostras do Docker para CodeBuild](#)

Mostra como usar imagens do Docker personalizadas, publicar imagens do Docker em um repositório no Amazon ECR e usar imagens do Docker em um registro privado.

[Saída de compilação hospedada em um bucket do S3](#)

Mostra como criar um site estático em um bucket do S3 usando os artefatos de compilação não criptografados.

[Exemplo de várias entradas e saídas](#)

Mostra como usar várias origens de entrada e vários artefatos de saída em um projeto de compilação.

[Amostras de execução paralela de testes](#)

Mostra como usar o comando `codebuild-tests-run` CLI para dividir e executar testes em ambientes de execução paralela.

[Versões do tempo de execução no exemplo de arquivo buildspec](#)

Mostra como especificar tempos de execução e suas versões no arquivo buildspec.

[Exemplo de versão de fonte](#)

Mostra como usar uma versão específica do seu código-fonte em um projeto de CodeBuild compilação.

[Amostras de repositórios de origem de terceiros para CodeBuild](#)

Mostra como criar BitBucket, GitHub Enterprise Server e GitHub pull requests usando webhooks usando. CodeBuild

[Defina os nomes dos artefatos no momento da compilação usando versionamento semântico](#)

Mostra como usar versionamento semântico para criar um nome de artefato no momento da compilação.

Amostras de serviços cruzados para CodeBuild

Você pode usar esses exemplos de serviços cruzados para fazer experiências com AWS CodeBuild:

[Exemplo do Amazon ECR](#)

Usa uma imagem do Docker em um repositório do Amazon ECR para usar o Apache Maven a fim de produzir um único arquivo JAR. As instruções de exemplo mostrarão como criar e enviar uma imagem do Docker para o Amazon ECR, criar um projeto Go, construir o projeto, executar o projeto e configurar permissões CodeBuild para permitir a conexão com o Amazon ECR.

[Exemplo do Amazon EFS](#)

Mostra como configurar um arquivo buildspec para que um CodeBuild projeto seja montado e construído em um sistema de arquivos Amazon EFS. As instruções de exemplo mostrarão como criar uma Amazon VPC, um sistema de arquivos na Amazon VPC, criar e compilar um projeto que usa a Amazon VPC e, em seguida, analisar o arquivo e as variáveis do projeto gerados.

[AWS CodePipeline amostras](#)

Mostra como usar AWS CodePipeline para criar uma compilação com compilações em lote, bem como várias fontes de entrada e vários artefatos de saída. Incluídos nesta seção estão exemplos de arquivos JSON que mostram estruturas de pipeline que criam compilações em lote com artefatos separados e artefatos combinados. É fornecido um exemplo adicional de JSON que mostra a estrutura do pipeline com várias origens de entrada e vários artefatos de saída.

[AWS Config amostra](#)

Mostra como configurar AWS Config. Lista quais CodeBuild recursos são monitorados e descreve como pesquisar CodeBuild projetos em AWS Config. Os exemplos de instruções mostrarão os pré-requisitos para a integração AWS Config, as etapas de configuração AWS Config e as etapas para pesquisar CodeBuild projetos e dados. AWS Config

[Exemplo de notificações de compilação](#)

Usa o Apache Maven para produzir um único arquivo JAR. Envia uma notificação de compilação aos assinantes de um tópico do Amazon SNS. Os exemplos de instruções mostram como configurar permissões para que eles CodeBuild possam se comunicar com o Amazon SNS e CloudWatch como criar e identificar CodeBuild tópicos no Amazon SNS, como inscrever destinatários no tópico e como configurar regras no CloudWatch

Exemplo de Amazon ECR para CodeBuild

Este exemplo usa uma imagem do Docker em um repositório de imagens do Amazon Elastic Container Registry (Amazon ECR) para compilar um exemplo do projeto Go.

Important

A execução dessa amostra pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por AWS CodeBuild AWS recursos e ações relacionados ao Amazon S3 AWS KMS, CloudWatch Logs e Amazon ECR. Para obter mais informações, consulte [CodeBuild preços, preços](#) do [Amazon S3, preços, preços](#) da Amazon e [AWS Key Management Service preços](#) do [Amazon CloudWatch Elastic Container Registry](#).

Tópicos

- [Execute o exemplo do Amazon ECR](#)

Execute o exemplo do Amazon ECR

Use as instruções a seguir para executar a amostra do Amazon ECR para CodeBuild.

Como executar este exemplo

1. Para criar e enviar por push uma imagem do Docker para o repositório de imagens no Amazon ECR, siga as etapas na seção [Execute o exemplo “Publicar imagem do Docker no Amazon ECR”](#) do [Exemplo “Publicar imagem do Docker no Amazon ECR”](#).
2. Criar um projeto Go

- a. Crie os arquivos conforme descrito nas [Arquivos do projeto Go](#) seções [Estrutura do projeto Go](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub

⚠ Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

- b. Crie um projeto de compilação, execute a compilação e exiba as respectivas informações.

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "sample-go-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```


- c. Para obter o artefato de saída de compilação, abra seu bucket de saída do S3.
 - d. Faça o download do arquivo *GoOutputArtifact*.zip para seu computador ou instância local. Extraia o conteúdo do arquivo. Nesse conteúdo, procure o arquivo hello.
3. Se uma das afirmações a seguir for verdadeira, você deverá adicionar permissões ao seu repositório de imagens no Amazon ECR para que ele AWS CodeBuild possa inserir sua imagem do Docker no ambiente de criação.
- Seu projeto usa CodeBuild credenciais para extrair imagens do Amazon ECR. Isso é indicado por um valor de CODEBUILD no atributo `imagePullCredentialsType` de seu `ProjectEnvironment`.
 - O projeto usa uma imagem do Amazon ECR entre contas. Nesse caso, o projeto deverá usar o perfil de serviço para extrair imagens do Amazon ECR. Para habilitar esse comportamento, defina o atributo `imagePullCredentialsType` de seu `ProjectEnvironment` como `SERVICE_ROLE`.
1. Abra o console do Amazon ECR em <https://console.aws.amazon.com/ecr/>.
 2. Na lista de nomes de repositório, escolha o nome do repositório que você criou ou selecionou.
 3. No painel de navegação, selecione Permissions (Permissões), Edit (Editar) e Add statement (Adicionar instrução).
 4. Em Statement name (Nome da instrução), digite um identificador (por exemplo, **CodeBuildAccess**).
 5. Em Effect (Efeito), deixe a opção Allow (Permitir) selecionada. Isso indica que você deseja permitir o acesso a outra conta da AWS.
 6. Em Principal, siga um dos seguintes procedimentos:
 - Se seu projeto usa CodeBuild credenciais para extrair uma imagem do Amazon ECR, em Service principal, insira. **codebuild.amazonaws.com**
 - Se o seu projeto usa uma imagem do Amazon ECR entre contas, como AWS conta IDs, insira IDs as AWS contas às quais você deseja conceder acesso.
 7. Ignore a lista All IAM entities.
 8. Em Ação, selecione as ações somente de extração: `ecr:GetDownloadUrlForLayer`, `ecr:` e `ecr:BatchGetImage`. `BatchCheckLayerAvailability`
 9. Em Condições, adicione o seguinte:

```

    "StringEquals":{
      "aws:SourceAccount":"<AWS-account-ID>",
      "aws:SourceArn":"arn:aws:codebuild:<region>:<AWS-account-ID>:project/<project-name>"
    }
  }
}

```

10 Escolha Salvar.

Essa política é exibida em Permissions (Permissões). O principal é a informação inserida em Principal na etapa 3 deste procedimento:

- Se seu projeto usa CodeBuild credenciais para extrair uma imagem do Amazon ECR, "codebuild.amazonaws.com" aparece em Princípios de serviço.
- Se o seu projeto usa uma imagem do Amazon ECR entre contas, o ID da AWS conta à qual você deseja conceder acesso aparecerá em Conta AWS . IDs

O exemplo de política a seguir usa CodeBuild credenciais e uma imagem do Amazon ECR entre contas.

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"CodeBuildAccessPrincipal",
      "Effect":"Allow",
      "Principal":{"
        "Service":"codebuild.amazonaws.com"
      }},
      "Action":[
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Condition":{"
        "StringEquals":{"
          "aws:SourceArn":"arn:aws:codebuild:<region>:<aws-account-id>:project/<project-name>",
          "aws:SourceAccount":"<aws-account-id>"
        }
      }
    }
  ],
  {

```

```

        "Sid": "CodeBuildAccessCrossAccount",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::<AWS-account-ID>:root"
        },
        "Action": [
            "ecr:GetDownloadUrlForLayer",
            "ecr:BatchGetImage",
            "ecr:BatchCheckLayerAvailability"
        ]
    }
}

```

- Se seus projetos usam CodeBuild credenciais e você gostaria que eles tivessem acesso aberto ao repositório Amazon ECR, você pode omitir `Condition` as chaves e adicionar o seguinte exemplo de política. CodeBuild

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<AWS-account-ID>:root"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    }
  ]
}

```

```
    ]
  }
]
}
```

4. Crie um projeto de compilação, execute a compilação e exiba suas informações.

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "amazon-ecr-sample-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "account-ID.dkr.ecr.region-ID.amazonaws.com/your-Amazon-ECR-repo-name:tag",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

5. Para obter o artefato de saída de compilação, abra seu bucket de saída do S3.
6. Faça o download do arquivo *GoOutputArtifact*.zip para seu computador ou instância local. Extraia o conteúdo do arquivo *GoOutputArtifact*.zip. Nesse conteúdo, procure o arquivo `hello`.

Estrutura do projeto Go

Essa amostra pressupõe esta estrutura de diretório.

(root directory name)

```
### buildspec.yml
```

```
### hello.go
```

Arquivos do projeto Go

Esta amostra usa estes arquivos.

buildspec.yml (em *(root directory name)*)

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
  build:
    commands:
      - echo Build started on `date`
      - echo Compiling the Go code
      - go build hello.go
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - hello
```

hello.go (em *(root directory name)*)

```
package main
import "fmt"

func main() {
    fmt.Println("hello world")
    fmt.Println("1+1 =", 1+1)
    fmt.Println("7.0/3.0 =", 7.0/3.0)
    fmt.Println(true && false)
    fmt.Println(true || false)
    fmt.Println(!true)
}
```

Exemplo do Amazon Elastic File System para AWS CodeBuild

Talvez você queira criar suas AWS CodeBuild versões no Amazon Elastic File System, um serviço de arquivos compartilhado e escalável para instâncias da Amazon EC2 . A capacidade de armazenamento com o Amazon EFS é elástica, portanto, aumenta ou diminui à medida que arquivos são adicionados e removidos. Ela tem uma interface simples de serviços da Web que pode ser usada para criar e configurar sistemas de arquivos. Ela também gerencia toda a infraestrutura de armazenamento de arquivos para você, para que você não precise se preocupar com a implantação, atualização nem com a manutenção das configurações do sistema de arquivos. Para obter mais informações, consulte [What is Amazon Elastic File System?](#) no Guia do usuário do Amazon Elastic File System.

Este exemplo mostra como configurar um CodeBuild projeto para que ele monte e, em seguida, crie um aplicativo Java em um sistema de arquivos Amazon EFS. Antes de começar, você deve ter um aplicativo Java pronto para ser construído e carregado em um bucket de entrada do S3 ou em um repositório AWS CodeCommit GitHub,, GitHub Enterprise Server ou Bitbucket.

Os dados em trânsito para o sistema de arquivos estão criptografados. Para criptografar os dados em trânsito usando uma imagem diferente, consulte [Criptografar dados em trânsito](#).

Tópicos

- [Use AWS CodeBuild com o Amazon Elastic File System](#)
- [Solucionar problemas de integração do Amazon EFS](#)

Use AWS CodeBuild com o Amazon Elastic File System

A amostra abrange as quatro etapas de alto nível necessárias para usar o Amazon EFS com AWS CodeBuild. Eles são:

1. Crie uma nuvem privada virtual (VPC) em sua AWS conta.
2. Crie um sistema de arquivos que usa essa VPC.
3. Crie e construa um CodeBuild projeto que use a VPC. O CodeBuild projeto usa o seguinte para identificar o sistema de arquivos:
 - Um identificador de sistema de arquivos exclusivo. Você escolhe o identificador ao especificar o sistema de arquivos no projeto de compilação.
 - O ID do sistema de arquivos. O ID é exibido quando você visualiza o sistema de arquivos no console do Amazon EFS.

- Um ponto de montagem. Este é um diretório no contêiner do Docker que monta o sistema de arquivos.
 - Opções de montagem. Elas incluem os detalhes sobre como montar o sistema de arquivos.
4. Revise o projeto de compilação para garantir que os arquivos e variáveis corretos do projeto foram gerados.

 Note

Um sistema de arquivos criado no Amazon EFS é compatível somente com as plataformas Linux.

Tópicos

- [Etapa 1: criar uma VPC usando AWS CloudFormation](#)
- [Etapa 2: criar um sistema de arquivos do Amazon Elastic File System com a VPC](#)
- [Etapa 3: criar um CodeBuild projeto para usar com o Amazon EFS](#)
- [Etapa 4: revisar o projeto de compilação](#)

Etapa 1: criar uma VPC usando AWS CloudFormation

Crie sua VPC com um AWS CloudFormation modelo.

1. Siga as instruções de uso [AWS CloudFormation Modelo de VPC](#) AWS CloudFormation para criar uma VPC.

 Note

A VPC criada por esse AWS CloudFormation modelo tem duas sub-redes privadas e duas sub-redes públicas. Use somente as sub-redes privadas quando usar o AWS CodeBuild para montar o sistema de arquivos criado no Amazon EFS. Se você usar uma das sub-redes públicas, a compilação falhará.

2. Faça login no AWS Management Console e abra o console da Amazon VPC em. <https://console.aws.amazon.com/vpc/>
3. Escolha a VPC com a qual você criou. AWS CloudFormation

4. Na guia Description (Descrição) anote o nome da sua VPC e do ID. Os dois serão necessários quando você criar o projeto do AWS CodeBuild posteriormente neste exemplo.

Etapa 2: criar um sistema de arquivos do Amazon Elastic File System com a VPC

Crie um sistema de arquivos Amazon EFS simples para esse exemplo usando a VPC criada anteriormente.

1. Faça login no AWS Management Console e abra o console do Amazon EFS em <https://console.aws.amazon.com/efs/>.
2. Escolha Create file system (Criar sistema de arquivos).
3. Em VPC, escolha o nome da VPC que você anotou anteriormente neste exemplo.
4. Deixe as zonas de disponibilidade associadas às suas sub-redes selecionadas.
5. Escolha Próxima etapa.
6. Em Adicionar tags, para a chave Nome, em Valor, insira o nome do sistema de arquivos Amazon EFS.
7. Mantenha Bursting e General Purpose (Uso geral) selecionados como modos padrão de desempenho e de taxa de transferência e selecione Next Step (Próxima etapa).
8. Para Configure client access (Configurar acesso de cliente), escolha Next Step (Próxima etapa).
9. Escolha Criar sistema de arquivos.
10. (Opcional) Recomendamos adicionar uma política ao sistema de arquivos Amazon EFS que imponha a criptografia de dados em trânsito. No console do Amazon EFS, escolha Política do sistema de arquivos, escolha Editar, selecione a caixa Aplicar criptografia em trânsito para todos os clientes e escolha Salvar.

Etapa 3: criar um CodeBuild projeto para usar com o Amazon EFS

Crie um AWS CodeBuild projeto que use a VPC que você criou anteriormente neste exemplo. Quando a compilação for executada, ela montará o sistema de arquivos do Amazon EFS criado anteriormente. Depois, armazenará o arquivo .jar criado pelo aplicativo Java no diretório do ponto de montagem do sistema de arquivos.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, digite um nome para o seu projeto.

4. Em Provedor de origem, escolha o repositório que contém o aplicativo Java que você deseja criar.
5. Insira informações, como uma URL do repositório, que CodeBuild usa para localizar seu aplicativo. As opções são diferentes para cada provedor de origem. Para obter mais informações, consulte [Choose source provider](#).
6. Em Environment image (Imagem do ambiente), escolha Managed image (Imagem gerenciada).
7. Em Operating system (Sistema operacional), escolha Amazon Linux 2.
8. Em Runtime(s) (Tempo(s) de execução), selecione Standard (Padrão).
9. Em Imagem, escolha aws/codebuild/amazonlinux-x86_64-standard:4.0.
10. Em Environment type (Tipo de ambiente), escolha Linux.
11. Em Função do serviço, selecione Nova função de serviço. Em Nome da função, insira um nome para a função CodeBuild criada para você.
12. Expanda Configuração Adicional.
13. Selecione Enable this flag if you want to build Docker images or want your builds to get elevated privileges (Ativar este sinalizador se quiser criar imagens do Docker ou desejar que suas compilações obtenham privilégios elevados).

 Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

14. Em VPC, escolha o ID do VPC.
15. Em Sub-redes, escolha uma ou mais sub-redes privadas associadas à VPC. É necessário usar sub-redes privadas em uma compilação que monta um sistema de arquivos Amazon EFS. Se você usar uma sub-rede pública, a compilação falhará.
16. Em Security groups (Grupos de segurança), escolha o grupo de segurança padrão.
17. Em File systems (Sistemas de arquivo), insira as seguintes informações:
 - Em Identifier (Identificador), insira um identificador exclusivo do sistema de arquivos. Deve ter menos de 129 caracteres e conter apenas caracteres alfanuméricos e sublinhados. O CodeBuild usa esse identificador para criar uma variável de ambiente que identifica o sistema de arquivos elástico. O formato da variável de ambiente é

CODEBUILD_<file_system_identifier> em letras maiúsculas. Por exemplo, se você inserir my_efs, a variável de ambiente será CODEBUILD_MY_EFS.

- Para ID, escolha o ID do sistema de arquivos.
- (Opcional) Insira um diretório no sistema de arquivos. CodeBuild monta esse diretório. Se você deixar o Directory path (Caminho do diretório) em branco, o CodeBuild monta todo o sistema de arquivos. O caminho é relativo à raiz do sistema de arquivos.
- Em Ponto de montagem, digite o caminho absoluto do diretório no contêiner de compilação onde o sistema de arquivos é montado. Se esse diretório não existir, CodeBuild cria-o durante a compilação.
- (Opcional) Insira as opções de montagem. Se você deixar as opções de montagem em branco, CodeBuild usa suas opções de montagem padrão:

```
nfsvers=4.1
rsize=1048576
wsize=1048576
hard
timeo=600
retrans=2
```

Para obter mais informações, consulte [Recommended NFS Mount Options](#) no Guia do usuário do Amazon Elastic File System.

18. Em Build specification (Criar especificação), escolha Insert build commands (Inserir comandos de compilação) e escolha Switch to editor (Alternar para editor).
19. Insira os comandos buildspec a seguir no editor. Substitua <file_system_identifier> pelo identificador inserido na etapa 17. Use letras maiúsculas (por exemplo, CODEBUILD_MY_EFS).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
  build:
    commands:
      - mvn compile -Dpgp.skip=true -Dmaven.repo.local=
        $CODEBUILD_<file_system_identifier>
```

20. Use os valores padrão para todas as outras configurações e, em seguida, escolha Criar projeto de compilação. Quando a compilação estiver concluída, a página do console para o projeto será exibida.
21. Selecione Iniciar compilação.

Etapa 4: revisar o projeto de compilação

Depois que seu AWS CodeBuild projeto for construído:

- Você terá um arquivo .jar criado pela aplicação Java criada para o sistema de arquivos Amazon EFS no diretório do ponto de montagem.
- Uma variável de ambiente que identifica o sistema de arquivos é criada usando o identificador de sistema de arquivos inserido quando o projeto foi criado.

Para obter mais informações, consulte [Mounting file systems](#) no Guia do usuário do Amazon Elastic File System.

Solucionar problemas de integração do Amazon EFS

A seguir estão os erros que você pode encontrar ao configurar o Amazon EFS com CodeBuild.

Tópicos

- [CLIENT_ERROR: falha na montagem de "127.0.0.1: /". Permissão negada](#)
- [CLIENT_ERROR: falha na montagem de "127.0.0.1: /". Conexão redefinida pelo par](#)
- [VPC_CLIENT_ERROR: Erro inesperado: EC2 UnauthorizedOperation](#)

CLIENT_ERROR: falha na montagem de "127.0.0.1: /". Permissão negada

A autorização do IAM não é suportada para montar o Amazon EFS com CodeBuild. Se você estiver usando uma política personalizada do sistema de arquivos do Amazon EFS, precisará conceder acesso de leitura e gravação a todas as entidades principais do IAM. Por exemplo:

```
"Principal": {  
  "AWS": "*"   
}
```

CLIENT_ERROR: falha na montagem de “127.0.0.1: /”. Conexão redefinida pelo par

Há duas causas possíveis para esse erro:

- A sub-rede CodeBuild VPC está em uma zona de disponibilidade diferente do destino de montagem do Amazon EFS. É possível resolver isso incluindo uma sub-rede da VPC na mesma zona de disponibilidade do destino de montagem do Amazon EFS.
- O grupo de segurança não tem permissões para se comunicar com o Amazon EFS. É possível resolver isso adicionando uma regra de entrada para viabilizar todo o tráfego da VPC (adicione o bloco CIDR primário da VPC) ou do próprio grupo de segurança.

VPC_CLIENT_ERROR: Erro inesperado: EC2 UnauthorizedOperation

Esse erro ocorre quando todas as sub-redes em sua configuração de VPC para o CodeBuild projeto são sub-redes públicas. É necessário ter, pelo menos, uma sub-rede privada na VPC para garantir a conectividade de rede.

AWS CodePipeline amostras para CodeBuild

Esta seção descreve exemplos de integrações entre CodePipeline e CodeBuild

Amostra	Descrição
Amostras de CodeBuild integrações CodePipeline /e compilações em lote	Esses exemplos demonstram como usar AWS CodePipeline para criar um projeto de compilação que usa compilações em lote.
Amostra de uma CodeBuild integração CodePipeline/com várias fontes de entrada e artefatos de saída	Este exemplo demonstra como usar AWS CodePipeline para criar um projeto de construção que usa várias fontes de entrada para criar vários artefatos de saída.

Amostras de CodeBuild integrações CodePipeline /e compilações em lote

AWS CodeBuild suporta compilações em lote. Os exemplos a seguir demonstram como usar AWS CodePipeline para criar um projeto de compilação que usa compilações em lote.

Você pode usar um arquivo formatado em JSON que define a estrutura do seu pipeline e, em seguida, usá-lo com o AWS CLI para criar o pipeline. Para obter mais informações, consulte [AWS CodePipeline Pipeline structure reference](#) no Guia do usuário do AWS CodePipeline .

Compilação em lote com artefatos individuais

Use o arquivo JSON a seguir como exemplo de uma estrutura de pipeline que cria uma compilação em lote com artefatos separados. Para habilitar compilações em lote CodePipeline, defina o `BatchEnabled` parâmetro do `configuration` objeto como `true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "<my-input-bucket-name>",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          },
          {
            "inputArtifacts": [],
            "name": "Source2",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
```

```
        "version": "1",
        "provider": "S3"
    },
    "outputArtifacts": [
        {
            "name": "source2"
        }
    ],
    "configuration": {
        "S3Bucket": "<my-other-input-bucket-name>",
        "S3ObjectKey": "my-other-source-code-file-name.zip"
    },
    "runOrder": 1
}
]
},
{
    "name": "Build",
    "actions": [
        {
            "inputArtifacts": [
                {
                    "name": "source1"
                },
                {
                    "name": "source2"
                }
            ],
            "name": "Build",
            "actionTypeId": {
                "category": "Build",
                "owner": "AWS",
                "version": "1",
                "provider": "CodeBuild"
            },
            "outputArtifacts": [
                {
                    "name": "build1"
                },
                {
                    "name": "build1_artifact1"
                },
                {
                    "name": "build1_artifact2"
                }
            ]
        }
    ]
}
```

```

        },
        {
            "name": "build2_artifact1"
        },
        {
            "name": "build2_artifact2"
        }
    ],
    "configuration": {
        "ProjectName": "my-build-project-name",
        "PrimarySource": "source1",
        "BatchEnabled": "true"
    },
    "runOrder": 1
}
]
}
],
"artifactStore": {
    "type": "S3",
    "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}

```

Veja a seguir um exemplo de um arquivo CodeBuild buildspec que funcionará com essa configuração de pipeline.

```

version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:

```

```

- echo 'file' > output_file

artifacts:
  files:
    - output_file
  secondary-artifacts:
    artifact1:
      files:
        - output_file
    artifact2:
      files:
        - output_file

```

Os nomes dos artefatos de saída especificados no arquivo JSON do pipeline devem coincidir com o identificador das compilações e dos artefatos definidos no arquivo buildspec. A sintaxe é *buildIdentifier* para os artefatos primários e *buildIdentifier_artifactIdentifier* para os artefatos secundários.

Por exemplo, para o nome do artefato de saída build1, CodeBuild fará o upload do artefato primário de build1 para o local de. build1 Para o nome de saída build1_artifact1, CodeBuild fará o upload do artefato secundário artifact1 de build1 para o local de build1_artifact1, e assim por diante. Se somente um local de saída for especificado, o nome deverá ser *buildIdentifier* somente.

Depois que você criar o arquivo JSON, poderá criar o pipeline. Use o AWS CLI para executar o comando `create-pipeline` e passar o arquivo para o parâmetro. `--cli-input-json` Para obter mais informações, consulte [Create a pipeline \(CLI\)](#) no Guia do usuário do AWS CodePipeline .

Compilação em lote com artefatos combinados

Use o arquivo JSON a seguir como exemplo de uma estrutura de pipeline que cria uma compilação em lote com artefatos combinados. Para habilitar compilações em lote CodePipeline, defina o `BatchEnabled` parâmetro do `configuration` objeto como `true`. Para combinar os artefatos de compilação no mesmo local, defina o parâmetro `CombineArtifacts` do objeto `configuration` como `true`.

```

{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {

```



```
"name": "Source",
"actions": [
  {
    "inputArtifacts": [],
    "name": "Source1",
    "actionTypeId": {
      "category": "Source",
      "owner": "AWS",
      "version": "1",
      "provider": "S3"
    },
    "outputArtifacts": [
      {
        "name": "source1"
      }
    ],
    "configuration": {
      "S3Bucket": "<my-input-bucket-name>",
      "S3objectKey": "my-source-code-file-name.zip"
    },
    "runOrder": 1
  },
  {
    "inputArtifacts": [],
    "name": "Source2",
    "actionTypeId": {
      "category": "Source",
      "owner": "AWS",
      "version": "1",
      "provider": "S3"
    },
    "outputArtifacts": [
      {
        "name": "source2"
      }
    ],
    "configuration": {
      "S3Bucket": "<my-other-input-bucket-name>",
      "S3objectKey": "my-other-source-code-file-name.zip"
    },
    "runOrder": 1
  }
]
},
```

```
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "output1 "
        }
      ],
      "configuration": {
        "ProjectName": "my-build-project-name",
        "PrimarySource": "source1",
        "BatchEnabled": "true",
        "CombineArtifacts": "true"
      },
      "runOrder": 1
    }
  ]
},
{
  "artifactStore": {
    "type": "S3",
    "location": "<AWS-CodePipeline-internal-bucket-name>"
  },
  "name": "my-pipeline-name",
  "version": 1
}
```

Veja a seguir um exemplo de um arquivo CodeBuild buildspec que funcionará com essa configuração de pipeline.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
```

Se os artefatos combinados estiverem habilitados para a construção em lote, haverá somente uma saída permitida. CodeBuild combinará os artefatos primários de todas as compilações em um único arquivo ZIP.

Depois que você criar o arquivo JSON, poderá criar o pipeline. Use o AWS CLI para executar o comando `create-pipeline` e passar o arquivo para o parâmetro. `--cli-input-json` Para obter mais informações, consulte [Criar um pipeline \(CLI\)](#) no Guia do usuário do AWS CodePipeline .

Amostra de uma CodeBuild integração CodePipeline/com várias fontes de entrada e artefatos de saída

Um AWS CodeBuild projeto pode usar mais de uma fonte de entrada. Ele também pode criar mais de um artefato de saída. Este exemplo demonstra como usar AWS CodePipeline para criar um projeto de construção que usa várias fontes de entrada para criar vários artefatos de saída. Para obter mais informações, consulte [Várias fontes de entrada e exemplos de artefatos de saída](#).

Você pode usar um arquivo formatado em JSON que define a estrutura do seu pipeline e, em seguida, usá-lo com o AWS CLI para criar o pipeline. Use o seguinte arquivo JSON como um exemplo de estrutura de pipeline que cria uma compilação com mais de uma origem de entrada e

mais de um artefato de saída. Mais adiante neste exemplo, você verá como este arquivo especifica as várias entradas e saídas. Para obter mais informações, consulte a [referência da estrutura do CodePipeline pipeline](#) no Guia AWS CodePipeline do usuário.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "my-input-bucket-name",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          },
          {
            "inputArtifacts": [],
            "name": "Source2",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source2"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "configuration": {
    "S3Bucket": "my-other-input-bucket-name",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "AWS CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "artifact1"
        },
        {
          "name": "artifact2"
        }
      ],
      "configuration": {
        "ProjectName": "my-build-project-name",
        "PrimarySource": "source1"
      },
      "runOrder": 1
    }
  ]
}
```

```
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "AWS-CodePipeline-internal-bucket-name"
  },
  "name": "my-pipeline-name",
  "version": 1
}
```

Neste arquivo JSON:

- Uma das suas origens de entrada deve ser designada como `PrimarySource`. Essa fonte é o diretório em que CodeBuild procura e executa seu arquivo `buildspec`. A palavra-chave `PrimarySource` é usada para especificar a fonte primária na `configuration` seção do CodeBuild estágio no arquivo JSON.
- Cada origem de entrada é instalada em seu próprio diretório. Esse diretório é armazenado na variável de ambiente interna `$CODEBUILD_SRC_DIR` para a origem principal e `$CODEBUILD_SRC_DIR_yourInputArtifactName` para todas as outras origens. Para o pipeline neste exemplo, os dois diretórios da origem de entrada são `$CODEBUILD_SRC_DIR` e `$CODEBUILD_SRC_DIR_source2`. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).
- Os nomes dos artefatos de saída especificados no arquivo JSON do pipeline devem coincidir com os nomes dos artefatos secundários definidos no arquivo `buildspec`. Esse pipeline usa o arquivo `buildspec` a seguir: Para obter mais informações, consulte [Sintaxe de buildspec](#).

```
version: 0.2

phases:
  build:
    commands:
      - touch source1_file
      - cd $CODEBUILD_SRC_DIR_source2
      - touch source2_file

artifacts:
  files:
    - '**/*'
```

```
secondary-artifacts:
  artifact1:
    base-directory: $CODEBUILD_SRC_DIR
    files:
      - source1_file
  artifact2:
    base-directory: $CODEBUILD_SRC_DIR_source2
    files:
      - source2_file
```

Depois que você criar o arquivo JSON, poderá criar o pipeline. Use o AWS CLI para executar o comando `create-pipeline` e passar o arquivo para o parâmetro. `--cli-input-json` Para obter mais informações, consulte [Criar um pipeline \(CLI\)](#) no Guia do usuário do AWS CodePipeline .

AWS Config amostra com CodeBuild

AWS Config fornece um inventário de seus AWS recursos e um histórico das alterações de configuração desses recursos. AWS Config agora oferece suporte AWS CodeBuild como um AWS recurso, o que significa que o serviço pode rastrear seus CodeBuild projetos. Para obter mais informações sobre AWS Config, consulte [O que é AWS Config?](#) no Guia do AWS Config desenvolvedor.

Você pode ver as seguintes informações sobre CodeBuild recursos na página Inventário de Recursos no AWS Config console:

- Uma linha do tempo de suas alterações de CodeBuild configuração.
- Detalhes de configuração para cada CodeBuild projeto.
- Relacionamentos com outros AWS recursos.
- Uma lista de mudanças em seus CodeBuild projetos.

Tópicos

- [Use CodeBuild com AWS Config](#)
- [Etapa 3: Exibir AWS CodeBuild dados no AWS Config console](#)

Use CodeBuild com AWS Config

Os procedimentos deste tópico mostram como configurar AWS Config e pesquisar CodeBuild projetos.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: configurar AWS Config](#)
- [Etapa 2: pesquisar AWS CodeBuild projetos](#)

Pré-requisitos

Crie seu AWS CodeBuild projeto. Para obter instruções, consulte [Criar um projeto de compilação](#).

Etapa 1: configurar AWS Config

- [Configurar o AWS Config \(console\)](#)
- [Configuração do AWS Config \(AWS CLI\)](#)

Note

Depois de concluir a configuração, pode levar até 10 minutos para que você possa ver os AWS CodeBuild projetos no AWS Config console.

Etapa 2: pesquisar AWS CodeBuild projetos

1. Faça login no AWS Management Console e abra o AWS Config console em <https://console.aws.amazon.com/config>.
2. Na página Inventário de recursos, selecione AWS CodeBuild Projeto em Tipo de recurso. Role para baixo e marque a caixa de seleção CodeBuilddo projeto.
3. Escolha Look up.
4. Depois que a lista de CodeBuild projetos for adicionada, escolha o link do nome do CodeBuild projeto na coluna Config timeline.

Etapa 3: Exibir AWS CodeBuild dados no AWS Config console

Ao pesquisar recursos na página Inventário de recursos, você pode escolher o AWS Config cronograma para ver detalhes sobre seu CodeBuild projeto. A página de detalhes de um recurso fornece informações sobre a configuração, os relacionamentos e o número de alterações feitas nesse recurso.

Os blocos na parte superior da página são coletivamente chamados de cronograma. A linha do tempo mostra a data e a hora em que o registro foi feito.

Para obter mais informações, consulte [Visualização dos detalhes da configuração no AWS Config console](#) no Guia do AWS Config desenvolvedor.

Crie uma amostra de notificações para CodeBuild

O Amazon CloudWatch Events tem suporte integrado para AWS CodeBuild. CloudWatch Eventos é um fluxo de eventos do sistema que descreve as mudanças em seus AWS recursos. Com CloudWatch os Eventos, você escreve regras declarativas para associar eventos de interesse às ações automatizadas a serem tomadas. Este exemplo usa o Amazon CloudWatch Events e o Amazon Simple Notification Service (Amazon SNS) para enviar notificações de compilação aos assinantes sempre que as compilações forem bem-sucedidas, falharem, passarem de uma fase de construção para outra ou qualquer combinação desses eventos.

Important

A execução dessa amostra pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados à Amazon CloudWatch e ao Amazon SNS. Para obter mais informações, consulte [CodeBuild preços](#), [CloudWatch preços da Amazon](#) e preços do [Amazon SNS](#).

Tópicos

- [Execute o exemplo de notificações de compilação](#)
- [Referência do formato de entrada das notificações de compilação](#)

Execute o exemplo de notificações de compilação

Use o procedimento a seguir para executar o exemplo de notificações de compilação.

Como executar este exemplo

1. Se você já tem uma configuração de tópicos e assinou o Amazon SNS que deseja usar para este exemplo, vá para a etapa 4. Caso contrário, se você estiver usando um usuário do IAM em vez de uma conta AWS raiz ou um usuário administrador para trabalhar com o Amazon SNS, adicione a seguinte declaração (entre **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) ao usuário (ou grupo do IAM ao qual o usuário está associado). Não é recomendável usar uma conta AWS root. Essa declaração permite a visualização, a criação, a assinatura e o teste do envio de notificações a tópicos no Amazon SNS. As elipses (...) são usadas para agilizar e para ajudá-lo a encontrar onde adicionar a declaração. Não remova nenhuma declaração e não digite essas elipses na política existente.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

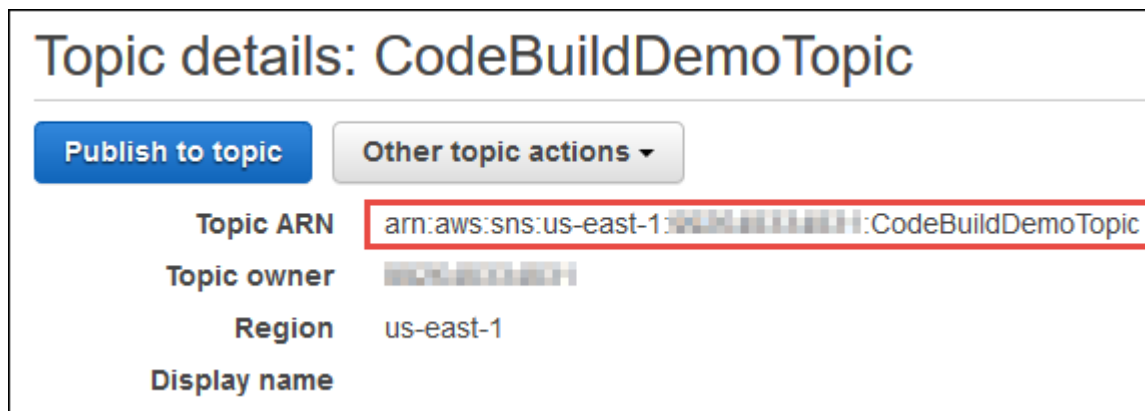
A entidade do IAM que modifica essa política deve ter permissão no IAM para alterar políticas.

Para obter mais informações, consulte [Editing customer managed policies](#) ou a seção “To edit or delete an inline policy for a group, user, or role” em [Working with inline policies \(console\)](#) no Guia do usuário do IAM.

2. Crie ou identifique um tópico no Amazon SNS. AWS CodeBuild usa CloudWatch Eventos para enviar notificações de criação para esse tópico por meio do Amazon SNS.

Para criar um tópico:

1. [Abra o console do Amazon SNS em https://console.aws.amazon.com/sns](https://console.aws.amazon.com/sns).
2. Escolha Criar tópico.
3. Em Create new topic (Criar novo tópico), em Topic name (Nome do tópico), insira um nome para o tópico (por exemplo, **CodeBuildDemoTopic**). (Se escolher um nome diferente, substitua-o ao longo desse exemplo.)
4. Escolha Criar tópico.
5. Na CodeBuildDemoTopic página Detalhes do tópico:, copie o valor do ARN do tópico. Você precisa desse valor na próxima etapa.



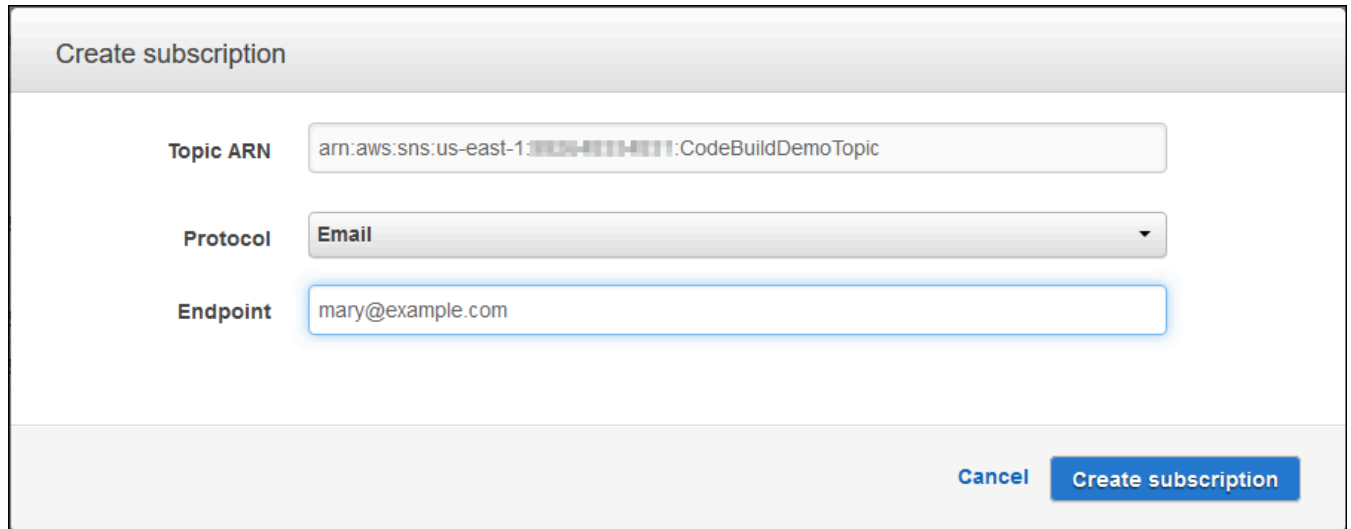
Para obter mais informações, consulte [Create a topic](#) no Guia do desenvolvedor do SNS.

3. Faça a assinatura de um ou mais destinatários para o tópico para que recebam notificações por e-mail.

Para fazer a assinatura de um destinatário para um tópico:

1. Com o console do Amazon SNS aberto na etapa anterior, no painel de navegação, selecione Assinaturas e Criar assinatura.

2. Em Create subscription (Criar assinatura), em Topic ARN (Tópico ARN), cole o ARN do tópico que você copiou da etapa anterior.
3. Em Protocolo, escolha E-mail.
4. Em Endpoint, insira o endereço de e-mail completo do destinatário.



Create subscription

Topic ARN

Protocol

Endpoint

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

5. Escolha Create Subscription.
6. O Amazon SNS envia um e-mail de confirmação de assinatura ao destinatário. Para começar a receber notificações por e-mail, o destinatário deve escolher o link Confirm subscription no e-mail de confirmação da assinatura. Assim que o destinatário clicar no link, se assinado com êxito, o Amazon SNS exibirá uma mensagem de confirmação no navegador web do destinatário.

Para obter mais informações, consulte [Subscribe to a topic](#) no Guia do desenvolvedor do Amazon SNS.

4. Se você estiver usando um usuário em vez de uma conta AWS raiz ou um usuário administrador para trabalhar com CloudWatch Eventos, adicione a seguinte declaração (entre **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) ao usuário (ou grupo do IAM ao qual o usuário está associado). Não é recomendável usar uma conta AWS root. Essa declaração é usada para permitir que o usuário trabalhe com CloudWatch Eventos. As elipses (. . .) são usadas para agilizar e para ajudá-lo a encontrar onde adicionar a declaração. Não remova nenhuma declaração e não digite essas elipses na política existente.

```
{
```

```
"Statement": [  
  ### BEGIN ADDING STATEMENT HERE ###  
  {  
    "Action": [  
      "events:*",  
      "iam:PassRole"  
    ],  
    "Resource": "*",  
    "Effect": "Allow"  
  },  
  ### END ADDING STATEMENT HERE ###  
  ...  
],  
"Version": "2012-10-17"  
}
```

Note

A entidade do IAM que modifica essa política deve ter permissão no IAM para alterar políticas.

Para obter mais informações, consulte [Editing customer managed policies](#) ou a seção “To edit or delete an inline policy for a group, user, or role” em [Working with inline policies \(console\)](#) no Guia do usuário do IAM.

5. Crie uma regra em CloudWatch Eventos. Para fazer isso, abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch>.
6. No painel de navegação, em Events, escolha Rules e Create rule.
7. Na Step 1: Create rule page (página Etapa 1: criar regra), Event Pattern (Padrão de evento) e Build event pattern to match events by service (Criar padrão de eventos para corresponder a eventos por serviço) já devem estar selecionados.
8. Em Service Name (Nome do serviço), escolha CodeBuild. Em Event Type (Tipo de evento), All Events (Todos os eventos) já deve estar selecionado.
9. O código a seguir deve ser exibido na Event Pattern Preview (Visualização do padrão de evento):

```
{  
  "source": [  
    "aws.codebuild"  
  ]  
}
```

```
}
```

10. Escolha Edit (Editar) e substitua o código na Event Pattern Preview (Visualização do padrão de evento) por um dos dois padrões de regra a seguir.

Esse primeiro padrão de regra aciona um evento quando uma compilação é iniciada ou concluída, para os projetos de compilação especificados no AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build State Change"
  ],
  "detail": {
    "build-status": [
      "IN_PROGRESS",
      "SUCCEEDED",
      "FAILED",
      "STOPPED"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

Na regra anterior, faça as alterações no código a seguir conforme necessário.

- Para acionar um evento quando uma compilação é iniciada ou concluída, deixe todos os valores conforme mostrado na matriz `build-status` ou remova a matriz `build-status` junto.
- Para acionar um evento somente quando uma compilação for concluída, remova `IN_PROGRESS` da matriz `build-status`.
- Para acionar um evento somente quando uma compilação é iniciada, remova todos os valores, exceto `IN_PROGRESS` da matriz `build-status`.
- Para acionar eventos para todos os projetos de compilação, remova a matriz `project-name` junto.

- Para acionar eventos apenas para projetos de compilação individuais, especifique o nome de cada projeto de compilação na matriz `project-name`.

Esse segundo padrão de regra aciona um evento sempre que uma compilação passa de uma fase de compilação para outra, para os projetos de compilação especificados no AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build Phase Change"
  ],
  "detail": {
    "completed-phase": [
      "SUBMITTED",
      "PROVISIONING",
      "DOWNLOAD_SOURCE",
      "INSTALL",
      "PRE_BUILD",
      "BUILD",
      "POST_BUILD",
      "UPLOAD_ARTIFACTS",
      "FINALIZING"
    ],
    "completed-phase-status": [
      "TIMED_OUT",
      "STOPPED",
      "FAILED",
      "SUCCEEDED",
      "FAULT",
      "CLIENT_ERROR"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

Na regra anterior, faça as alterações no código a seguir conforme necessário.

- Para acionar um evento para cada alteração na fase de compilação (que pode enviar até nove notificações para cada compilação), deixe todos os valores conforme mostrado na matriz `completed-phase` ou remova a matriz `completed-phase` junto.
- Para acionar eventos apenas para alterações feitas na fase de compilação individuais, remova o nome de cada fase de compilação na matriz `completed-phase` para a qual você não deseja acionar um evento.
- Para acionar um evento para cada alteração feita no status da fase de compilação, deixe todos os valores conforme mostrado na matriz `completed-phase-status` ou remova a matriz `completed-phase-status` junto.
- Para acionar eventos apenas para alterações feitas no status da fase de compilação individuais, remova o nome de cada status da fase de compilação na matriz `completed-phase-status` para a qual você não deseja acionar um evento.
- Para acionar eventos para todos os projetos de compilação, remova a matriz `project-name`.
- Para acionar eventos para projetos de compilação individuais, especifique o nome de cada projeto de compilação na matriz `project-name`.

Para obter mais informações sobre padrões de eventos, consulte [Padrões de eventos](#) no Guia EventBridge do usuário da Amazon.

Para obter mais informações sobre a filtragem com padrões de eventos, consulte [Filtragem baseada em conteúdo com padrões de eventos no Guia](#) do usuário da Amazon EventBridge .

 Note

Se quiser acionar eventos para alterações feitas no estado de compilação e alterações na fase de compilação, é necessário criar duas regras à parte: uma para alterações feitas no estado da compilação e outra para alterações feitas na fase de compilação. Se você tentar integrar ambas as regras em uma única, a regra combinada poderá produzir resultados inesperados ou deixar de funcionar em conjunto.

Quando você tiver concluído a substituição do código, escolha Save.

11. Em Targets, escolha Add target.

12. Na lista de destinos, escolha SNS topic.
13. Em Topic, escolha o tópico que você identificou ou criou anteriormente.
14. Expanda Configure input e escolha Input Transformer.
15. Na caixa Input Path (Caminho de entrada), digite um dos caminhos de entrada a seguir.

Para uma regra com um valor detail-type de CodeBuild Build State Change, insira o seguinte.

```
{"build-id":"$.detail.build-id","project-name":"$.detail.project-name","build-status":"$.detail.build-status"}
```

Para uma regra com um valor detail-type de CodeBuild Build Phase Change, insira o seguinte.

```
{"build-id":"$.detail.build-id","project-name":"$.detail.project-name","completed-phase":"$.detail.completed-phase","completed-phase-status":"$.detail.completed-phase-status"}
```

Para obter outros tipos de informações, consulte [Referência do formato de entrada das notificações de compilação](#).

16. Na caixa Input Template (Modelo de entrada), insira um dos modelos de entrada.

Para uma regra com um valor detail-type de CodeBuild Build State Change, insira o seguinte.

```
"Build '<build-id>' for build project '<project-name>' has reached the build status of '<build-status>'."
```

Para uma regra com um valor detail-type de CodeBuild Build Phase Change, insira o seguinte.

```
"Build '<build-id>' for build project '<project-name>' has completed the build phase of '<completed-phase>' with a status of '<completed-phase-status>'."
```

17. Escolha Configure details (Configurar detalhes).
18. Na página Step 2: Configure rule details (Etapa 2: configurar detalhes da regra) insira um nome e uma descrição opcional. Em State (Estado), deixe Enabled (Habilitado) selecionado.

19. Escolha Criar regra.
20. Crie projetos de compilação, execute as compilações e exiba as informações de compilação.
21. Confirme se agora CodeBuild está enviando notificações de compilação com sucesso. Por exemplo, verifique se agora os e-mails de notificação de compilação estão em sua caixa de entrada.

Para alterar o comportamento de uma regra, no CloudWatch console, escolha a regra que você deseja alterar, escolha Ações e, em seguida, escolha Editar. Faça alterações na regra e escolha Configure details (Configurar detalhes) e Update rule (Atualizar regra).

Para parar de usar uma regra para enviar notificações de compilação, no CloudWatch console, escolha a regra que você deseja parar de usar, escolha Ações e, em seguida, escolha Desativar.

Para excluir completamente uma regra, no CloudWatch console, escolha a regra que você deseja excluir, escolha Ações e, em seguida, escolha Excluir.

Referência do formato de entrada das notificações de compilação

CloudWatch entrega notificações no formato JSON.

As notificações de alteração feita no estado da compilação usam o seguinte formato:

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "build-status": "SUCCEEDED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "additional-information": {
      "artifact": {
```

```

    "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
    "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
    "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
  },
  "environment": {
    "image": "aws/codebuild/standard:5.0",
    "privileged-mode": false,
    "compute-type": "BUILD_GENERAL1_SMALL",
    "type": "LINUX_CONTAINER",
    "environment-variables": []
  },
  "timeout-in-minutes": 60,
  "build-complete": true,
  "initiator": "MyCodeBuildDemoUser",
  "build-start-time": "Sep 1, 2017 4:12:29 PM",
  "source": {
    "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
    "type": "S3"
  },
  "logs": {
    "group-name": "/aws/codebuild/my-sample-project",
    "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
    "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
  },
  "phases": [
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:12:29 PM",
      "duration-in-seconds": 0,
      "phase-type": "SUBMITTED",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:13:05 PM",
      "duration-in-seconds": 36,
      "phase-type": "PROVISIONING",
      "phase-status": "SUCCEEDED"
    }
  ]
}

```

```
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:05 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 4,
  "phase-type": "DOWNLOAD_SOURCE",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
```

```

        "end-time": "Sep 1, 2017 4:14:21 PM",
        "duration-in-seconds": 0,
        "phase-type": "UPLOAD_ARTIFACTS",
        "phase-status": "SUCCEEDED"
    },
    {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:14:21 PM",
        "end-time": "Sep 1, 2017 4:14:26 PM",
        "duration-in-seconds": 4,
        "phase-type": "FINALIZING",
        "phase-status": "SUCCEEDED"
    },
    {
        "start-time": "Sep 1, 2017 4:14:26 PM",
        "phase-type": "COMPLETED"
    }
]
},
"current-phase": "COMPLETED",
"current-phase-context": "[]",
"version": "1"
}
}

```

As notificações de alteração feita na fase de compilação usam o seguinte formato:

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",

```

```
"build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
"completed-phase-context": "[]",
"additional-information": {
  "artifact": {
    "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
    "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
    "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
  },
  "environment": {
    "image": "aws/codebuild/standard:5.0",
    "privileged-mode": false,
    "compute-type": "BUILD_GENERAL1_SMALL",
    "type": "LINUX_CONTAINER",
    "environment-variables": []
  },
  "timeout-in-minutes": 60,
  "build-complete": true,
  "initiator": "MyCodeBuildDemoUser",
  "build-start-time": "Sep 1, 2017 4:12:29 PM",
  "source": {
    "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
    "type": "S3"
  },
  "logs": {
    "group-name": "/aws/codebuild/my-sample-project",
    "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
    "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
  },
  "phases": [
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:12:29 PM",
      "duration-in-seconds": 0,
      "phase-type": "SUBMITTED",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
```

```
    "start-time": "Sep 1, 2017 4:12:29 PM",
    "end-time": "Sep 1, 2017 4:13:05 PM",
    "duration-in-seconds": 36,
    "phase-type": "PROVISIONING",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:05 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 4,
    "phase-type": "DOWNLOAD_SOURCE",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "INSTALL",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "PRE_BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 70,
    "phase-type": "BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 0,
    "phase-type": "POST_BUILD",
```

```

        "phase-status": "SUCCEEDED"
    },
    {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:14:21 PM",
        "end-time": "Sep 1, 2017 4:14:21 PM",
        "duration-in-seconds": 0,
        "phase-type": "UPLOAD_ARTIFACTS",
        "phase-status": "SUCCEEDED"
    },
    {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:14:21 PM",
        "end-time": "Sep 1, 2017 4:14:26 PM",
        "duration-in-seconds": 4,
        "phase-type": "FINALIZING",
        "phase-status": "SUCCEEDED"
    },
    {
        "start-time": "Sep 1, 2017 4:14:26 PM",
        "phase-type": "COMPLETED"
    }
]
},
"completed-phase-status": "SUCCEEDED",
"completed-phase-duration-seconds": 4,
"version": "1",
"completed-phase-start": "Sep 1, 2017 4:14:21 PM",
"completed-phase-end": "Sep 1, 2017 4:14:26 PM"
}
}

```

Crie uma amostra de crachás com CodeBuild

AWS CodeBuild agora suporta o uso de emblemas de construção, que fornecem uma imagem incorporável e gerada dinamicamente (emblema) que exibe o status da versão mais recente de um projeto. Essa imagem pode ser acessada por meio de uma URL publicamente disponível gerada para seu CodeBuild projeto. Isso permite que qualquer pessoa visualize o status de um CodeBuild projeto. Como não contêm informações de segurança, os emblemas de compilação não exigem autenticação.

Tópicos

- [Criar um projeto de compilação com emblemas de compilação](#)
- [Acesse os emblemas de AWS CodeBuild construção](#)
- [Publique emblemas de CodeBuild construção](#)
- [CodeBuild status do crachá](#)

Criar um projeto de compilação com emblemas de compilação

Use um dos procedimentos a seguir para criar um projeto de compilação com emblemas de compilação ativados. Você pode usar AWS CLI ou AWS Management Console o.

Para criar um projeto de compilação com emblemas de compilação habilitados (AWS CLI)

- Para obter informações sobre como criar um projeto de compilação, consulte [Criar um projeto de compilação \(AWS CLI\)](#). Para incluir emblemas de construção em seu AWS CodeBuild projeto, você deve especificar *badgeEnabled* com um valor de `true`.

Para criar um projeto de compilação com emblemas de compilação habilitados (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.
4. Em Source (Origem), em Source provider (Fornecedor de origem), selecione tipo de fornecedor de código-fonte e faça um dos seguintes procedimentos:

Note

CodeBuild não oferece suporte à criação de emblemas com o provedor de origem do Amazon S3. Como AWS CodePipeline usa o Amazon S3 para transferências de artefatos, os emblemas de construção não são compatíveis com projetos de construção que fazem parte de um pipeline criado em. CodePipeline

- Se você tiver escolhido CodeCommit, em Repository (Repositório), escolha o nome do repositório. Selecione Habilitar emblema da compilação para que o status de compilação do projeto fique visível e possa ser incorporado.
- Se você escolher GitHub, siga as instruções para se conectar (ou reconectar) com GitHub. Na página GitHub Autorizar aplicativo, em Acesso à organização, escolha Solicitar acesso ao lado de cada repositório que você AWS CodeBuild deseja acessar. Depois que você escolher Authorize application (Autorizar aplicação), no console do AWS CodeBuild, em Repository (Repositório), escolha o nome do repositório que contém o código-fonte. Selecione Habilitar emblema da compilação para que o status de compilação do projeto fique visível e possa ser incorporado.
- Se você escolher Bitbucket, siga as instruções para se conectar (ou reconectar) ao Bitbucket. Na página Confirm access to your account do Bitbucket, em Organization access, escolha Grant access. Depois de escolher Conceder acesso, de volta ao AWS CodeBuild console, para Repositório, escolha o nome do repositório que contém o código-fonte. Selecione Habilitar emblema da compilação para que o status de compilação do projeto fique visível e possa ser incorporado.

 Important

A atualização da origem do projeto pode afetar a precisão dos emblemas de compilação do projeto.

5. Em Ambiente:

Para Imagem do ambiente, faça um dos seguintes procedimentos:

- Para usar uma imagem do Docker gerenciada por AWS CodeBuild, escolha Imagem gerenciada e, em seguida, faça seleções em Sistema operacional, Tempo de execução (s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.
- Para usar outra imagem do Docker, selecione Custom image (Imagem personalizada). Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS

- Para usar uma imagem privada do Docker, selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

6. Em Perfil de serviço, faça um dos seguintes procedimentos:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

 Note

Ao usar o console para criar ou atualizar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

7. Em Buildspec, siga um destes procedimentos:

- Escolha Usar um arquivo buildspec para usar o arquivo buildspec.yml no diretório raiz do código-fonte.
- Selecione Inserir comandos de compilação para usar o console para inserir comandos de compilação.

Para obter mais informações, consulte [Referência de buildspec](#).

8. Em Artefatos, para Tipo, faça um dos seguintes procedimentos:

- Se você não quiser criar artefatos de saída de compilação, selecione Nenhum artefato.
- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:

- Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. Por padrão, o nome do artefato é o nome do projeto. Se você quiser usar um nome diferente, insira-o na caixa de nome do artefato. Se você quiser gerar um arquivo ZIP, inclua a extensão zip.
 - Para Bucket name, selecione o nome do bucket de saída.
 - Se você tiver escolhido Inserir comandos de compilação anteriormente neste procedimento, em Arquivos de saída, insira os locais dos arquivos da compilação que deseja incluir na pasta ou no arquivo ZIP de saída da compilação. Para vários locais, separe-os com uma vírgula (por exemplo, `appspec.yml, target/my-app.jar`). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).
9. Expanda Configuração adicional e escolha as opções conforme apropriado.
 10. Selecione Create build project (Criar projeto de compilação). Na página Revisão, escolha Iniciar compilação para executar a compilação.

Acesse os emblemas de AWS CodeBuild construção

Você pode usar o AWS CodeBuild console ou o AWS CLI para acessar os emblemas de construção.

- No CodeBuild console, na lista de projetos de construção, na coluna Nome, escolha o link que corresponde ao projeto de construção. Na *project-name* página Criar projeto:, em Configuração, escolha Copiar URL do selo. Para obter mais informações, consulte [Visualizar detalhes de um projeto de compilação \(console\)](#).
- No AWS CLI, execute o `batch-get-projects` comando. O URL do emblema de compilação está incluído na seção de detalhes do ambiente do projeto da saída. Para obter mais informações, consulte [Visualizar detalhes de um projeto de compilação \(AWS CLI\)](#).

O URL de solicitação do emblema de compilação é gerado com uma ramificação padrão comum, mas é possível especificar qualquer ramificação no repositório de origem utilizado para executar uma compilação. Por exemplo:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&branch=<branch>
```

Você também pode especificar uma tag do repositório de origem substituindo o parâmetro `branch` pelo parâmetro `tag` no URL do emblema. Por exemplo:

```
https://codebuild.us-east-1.amazonaws.com/badges?uuid=...&tag=<tag>
```

Publique emblemas de CodeBuild construção

É possível exibir o status da versão mais recente em um arquivo de markdown usando o URL do emblema de compilação em uma imagem de markdown. Isso é útil para exibir o status da compilação mais recente no arquivo readme.md no seu repositório de origem (por exemplo, ou). GitHub CodeCommit Por exemplo:

```

```

CodeBuild status do crachá

O emblema de CodeBuild construção pode ter um dos seguintes status.

- **PASSING** A compilação mais recente na ramificação indicada passada.
- **FAILING** A compilação mais recente na ramificação indicada expirou, falhou, apresentou defeito ou foi parada.
- **IN_PROGRESS** A compilação mais recente na ramificação indicada está em andamento.
- **UNKNOWN** O projeto ainda não executou uma compilação para a ramificação indicada ou nenhuma. Além disso, o recurso de emblemas de compilação pode ter sido desabilitado.

'Relatório de teste usando a amostra AWS CLI'

Os testes que você especificar no arquivo buildspec são executados durante a compilação. Este exemplo mostra como usar o para AWS CLI incorporar testes em CodeBuild compilações. Você pode usar JUnit para criar testes de unidade ou usar outra ferramenta para criar testes de configuração. Depois, você pode avaliar os resultados do teste para corrigir problemas ou otimizar seu aplicativo.

Você pode usar a CodeBuild API ou o AWS CodeBuild console para acessar os resultados do teste. Este exemplo mostra como configurar seu relatório para que seus resultados de teste sejam exportados para um bucket do S3.

Tópicos

- [Executar o exemplo de relatório de teste](#)

Executar o exemplo de relatório de teste

Use as etapas a seguir para executar o exemplo de relatório de teste.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um grupo de relatórios](#)
- [Etapa 2: configurar um projeto com um grupo de relatórios](#)
- [Etapa 3: executar e visualizar resultados de um relatório](#)

Pré-requisitos

- Criar seus casos de teste. Este exemplo é escrito partindo-se do pressuposto de que você tem casos de teste para incluir em seu relatório de teste de amostra. Especifique o local dos arquivos de teste no arquivo buildspec.

Os seguintes formatos de arquivo de relatório de teste são compatíveis:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- Visual Studio TRX XML (.xml)

Crie seus casos de teste com qualquer estrutura de teste que possa criar arquivos de relatório em um desses formatos (por exemplo, JUnit plug-in Surefire, TestNG ou Cucumber).

- Crie um bucket do S3 e anote seu nome. Para obter mais informações, consulte [How do I create an S3 bucket?](#) no Guia do usuário do Amazon S3.
- Crie um perfil do IAM e anote o ARN. É necessário o ARN ao criar seu projeto de compilação.
- Se sua função não tiver as seguintes permissões, adicione-as.

```
{  
    "Effect": "Allow",  
    "Resource": [
```

```
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases"
  ]
}
```

Para obter mais informações, consulte [Permissões para operações de relatórios de testes](#).

Etapa 1: criar um grupo de relatórios

1. Crie um arquivo chamado `CreateReportGroupInput.json`.
2. Crie uma pasta no bucket do S3 onde os resultados do teste são exportados.
3. Copie o seguinte em `CreateReportGroupInput.json`. Em *<bucket-name>*, use o nome do bucket do S3. Em *<path-to-folder>*, insira o caminho para a pasta no bucket do S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "path": "<path-to-folder>",
      "packaging": "NONE"
    }
  }
}
```

4. Execute o comando a seguir no diretório que contenha `CreateReportGroupInput.json`.

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

A saída é semelhante à seguinte. Anote o ARN do `reportGroup`. Use-o ao criar um projeto que use esse grupo de relatórios.

```
{
  "reportGroup": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:report-group/<report-name>",
    "name": "<report-name>",
    "type": "TEST",
    "exportConfig": {
      "exportConfigType": "S3",
      "s3Destination": {
        "bucket": "<s3-bucket-name>",
        "path": "<folder-path>",
        "packaging": "NONE",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3"
      }
    },
    "created": 1570837165.885,
    "lastModified": 1570837165.885
  }
}
```

Etapa 2: configurar um projeto com um grupo de relatórios

Para executar um relatório, primeiro você cria um projeto de CodeBuild compilação configurado com seu grupo de relatórios. Os casos de teste especificados para o grupo de relatórios são executados quando você executa uma compilação.

1. Crie um arquivo buildspec chamado `buildspec.yml`.
2. Use o seguinte YAML como modelo para o seu arquivo `buildspec.yml`. Inclua os comandos que executam seus testes. Na seção `reports`, especifique os arquivos que contêm os resultados dos seus casos de teste. Esses arquivos armazenam os resultados do teste com os quais você pode acessar CodeBuild. Eles expiram 30 dias após sua criação. Esses arquivos são diferentes dos arquivos de resultado de caso de teste bruto exportados para um bucket do S3.

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: openjdk8
  build:
    commands:
```



```

- echo Running tests
- <enter commands to run your tests>

reports:
  <report-name-or-arn>: #test file information
  files:
    - '<test-result-files>'
  base-directory: '<optional-base-directory>'
  discard-paths: false #do not remove file paths from test result files

```

Note

Em vez do ARN de um grupo de relatórios existente, você também pode especificar um nome para um grupo de relatórios que não foi criado. Se você especificar um nome em vez de um ARN, CodeBuild cria um grupo de relatórios ao executar uma compilação. O nome dele contém o nome do projeto e o nome que você especificar no arquivo buildspec, neste formato: `project-name-report-group-name`. Para ter mais informações, consulte [Criar relatórios de teste](#) e [Nomenclatura do grupo de relatórios](#).

3. Crie um arquivo chamado `project.json`. Este arquivo contém entrada para o comando `create-project`.
4. Copie o seguinte JSON em `project.json`. Em `source`, insira o tipo e o local do repositório que contém os arquivos de origem. Em `serviceRole`, especifique o ARN da função que você está usando.

```

{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|
NO_SOURCE",
    "location": "<your-source-url>"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {

```

```
"type": "LINUX_CONTAINER",
"image": "aws/codebuild/standard:5.0",
"computeType": "small"
},
"serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-name>"
}
```

5. Execute o comando a seguir no diretório que contenha `project.json`. Isso cria um projeto chamado `test-project`.

```
aws codebuild create-project --cli-input-json file://project.json
```

Etapa 3: executar e visualizar resultados de um relatório

Nesta seção, você executa uma compilação do projeto criado anteriormente. Durante o processo de criação, CodeBuild cria um relatório com os resultados dos casos de teste. O relatório está contido no grupo de relatórios especificado.

1. Para iniciar uma compilação, execute o comando a seguir. `test-report-project` é o nome do projeto de compilação criado acima. Anote o ID de compilação que aparece na saída.

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

2. Execute o comando a seguir para obter informações sobre sua compilação, incluindo o ARN do seu relatório. Em `<build-id>`, especifique seu ID de compilação. Anote o ARN do relatório na propriedade `reportArns` da saída.

```
aws codebuild batch-get-builds --ids <build-id>
```

3. Execute o comando a seguir para obter detalhes sobre o relatório. Em `<report-arn>`, especifique o ARN do relatório.

```
aws codebuild batch-get-reports --report-arns <report-arn>
```

A saída é semelhante à seguinte. Essa saída de exemplo mostra quantos testes foram bem-sucedidos, falharam, foram ignorados, resultaram em erro ou retornaram um status desconhecido.

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<amzn-s3-demo-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      },
      "expired": 1575916770.0,
      "truncated": false,
      "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",
      "type": "TEST",
      "arn": "<report-arn>",
      "testSummary": {
        "durationInNanoSeconds": 6657770,
        "total": 11,
        "statusCounts": {
          "FAILED": 3,
          "SKIPPED": 7,
          "ERROR": 0,
          "SUCCEEDED": 1,
          "UNKNOWN": 0
        }
      }
    }
  ],
  "reportsNotFound": []
}
```

4. Execute o seguinte comando para listar informações sobre casos de teste para seu relatório. Em **<report-arn>**, especifique o ARN do seu relatório. No parâmetro opcional **--filter**, você pode especificar um resultado de status (SUCCEEDED, FAILED, SKIPPED, ERROR, ou UNKNOWN).

```
aws codebuild describe-test-cases \
  --report-arn <report-arn> \
  --filter status=SUCCEEDED|FAILED|SKIPPED|ERROR|UNKNOWN
```

A saída é semelhante à seguinte.

```
{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    },
    {
      "status": "SUCCEEDED",
      "name": "Test case 2",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    }
  ]
}
```

Amostras do Docker para CodeBuild

Esta seção descreve exemplos de integrações entre Docker e AWS CodeBuild

Amostra	Descrição
Docker em amostra de imagem personalizada para CodeBuild	Este exemplo cria e executa uma imagem do Docker usando CodeBuild uma imagem

Amostra	Descrição
	de compilação personalizada do Docker (docker:dind no Docker Hub).
Exemplo de servidor de criação de imagens Docker para CodeBuild	Esse exemplo transfere suas compilações do Docker para um servidor gerenciado de criação de imagens.
O Windows Docker cria uma amostra para CodeBuild	Este exemplo cria e executa uma imagem do Windows Docker usando CodeBuild.
Exemplo de “Publicar imagem do Docker em um repositório de imagens do Amazon ECR” para CodeBuild	Este exemplo produz como saída de compilação o de uma imagem do Docker e, depois, a envia a um repositório de imagens do Amazon Elastic Container Registry (Amazon ECR).
Registro privado com AWS Secrets Manager amostra para CodeBuild	Este exemplo mostra como usar uma imagem do Docker armazenada em um registro privado como seu ambiente CodeBuild de execução.

Docker em amostra de imagem personalizada para CodeBuild

O exemplo a seguir cria e executa uma imagem do Docker usando AWS CodeBuild uma imagem de compilação personalizada do Docker (docker:dind no Docker Hub).

Para saber como criar uma imagem do Docker usando uma imagem de construção fornecida CodeBuild com o suporte do Docker, consulte nosso [Exemplo “Publicar imagem do Docker no Amazon ECR”](#).

Important

A execução dessa amostra pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados ao Amazon S3 e CloudWatch aos AWS KMS registros. Para obter mais informações, consulte [CodeBuild preços](#), [preços](#) do [Amazon S3](#), [preços](#) e [AWS Key Management Service preços](#) da [Amazon CloudWatch](#).

Tópicos

- [Executar o Docker no exemplo de imagem personalizada](#)

Executar o Docker no exemplo de imagem personalizada

Use o procedimento a seguir para executar o Docker em um exemplo de imagem personalizada. Para obter mais informações sobre este exemplo, consulte [Docker em amostra de imagem personalizada para CodeBuild](#).

Para executar o Docker no exemplo de imagem personalizada

1. Crie os arquivos conforme descrito nas [Arquivos](#) seções [Estrutura de diretório](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

2. Crie um projeto de compilação, execute a compilação e exiba as respectivas informações.

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerCustomImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
}
```

```
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "docker:dind",
  "computeType": "BUILD_GENERAL1_SMALL",
  "privilegedMode": false
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

3. Para ver os resultados da compilação, procure no log da compilação a string Hello, World!. Para obter mais informações, consulte [Visualizar detalhes de compilação](#).

Estrutura de diretório

Essa amostra pressupõe esta estrutura de diretório.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Arquivos

A imagem de base do sistema operacional Ubuntu é usada nessa amostra. A amostra usa estes arquivos.

buildspec.yml (em (*root directory name*))

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
```

```
build:
  commands:
    - docker images
    - docker run helloworld echo "Hello, World!"
```

Dockerfile (em *(root directory name)*)

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Exemplo de servidor de criação de imagens Docker para CodeBuild

O exemplo a seguir transfere suas compilações do Docker para um servidor gerenciado de criação de imagens. Você pode adaptar esse exemplo para provisionar um servidor de criação de imagens Docker dedicado e gerenciado na configuração do seu CodeBuild projeto. Observe que a instância provisionada está ativa enquanto as compilações são executadas ativamente para o projeto, e a instância é interrompida quando as compilações não estão em execução. A instância provisionada é armazenada por até um mês antes de ser reciclada. Para obter mais informações, consulte [CodeBuild Docker Server Capability](#).

Important

A execução dessa amostra pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados ao Amazon S3 e CloudWatch aos AWS KMS registros. Para obter mais informações, consulte [CodeBuild preços](#), [preços](#) do [Amazon S3](#), [preços](#) e [AWS Key Management Service preços](#) da [Amazon CloudWatch](#).

Tópicos

- [Configurar um servidor Docker](#)

Configurar um servidor Docker

Use o procedimento a seguir para provisionar um ambiente de computação dedicado para um CodeBuild projeto que gerencia cargas de trabalho do Docker e armazena camadas de imagem do Docker.

Para configurar um servidor Docker

1. Crie os arquivos conforme descrito nas [Arquivos](#) seções [Estrutura de diretório](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

2. Crie um projeto de compilação, execute a compilação e visualize as informações relacionadas à compilação:
 - a. Na seção Ambiente do console, escolha Configuração adicional, navegue até Configuração do servidor Docker e selecione Habilitar servidor docker para este projeto. Em seguida, você pode escolher o tipo de computação do servidor Docker e fornecer uma credencial do Registro.
 - b. Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o create-project comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerServerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:5.0",
    "computeType": "BUILD_GENERAL1_LARGE",
    "dockerServer": [
```

```

        {
            "computeType": "BUILD_GENERAL1_LARGE",
            "securityGroupIds": [ "security-groups-ID" ]
        }
    ],
    "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}

```

Note

Os grupos de segurança configurados para servidores Docker devem permitir a entrada de tráfego de rede da VPC configurada no projeto. Eles devem permitir a entrada na porta 9876.

3. Para ver os resultados da compilação, procure no log da compilação a string Hello, World!. Para obter mais informações, consulte [Visualizar detalhes de compilação](#).

Estrutura de diretório

Essa amostra pressupõe esta estrutura de diretório.

```

(root directory name)
### buildspec.yml
### Dockerfile

```

Arquivos

A imagem de base do sistema operacional Ubuntu é usada nessa amostra. A amostra usa estes arquivos.

buildspec.yml (em (*root directory name*))

```

version: 0.2

phases:
  build:
    commands:
      - docker buildx build .
      - docker run helloworld echo "Hello, World!"

```

Dockerfile (em *(root directory name)*)

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

RUN echo "Hello World"
```

O Windows Docker cria uma amostra para CodeBuild

O exemplo a seguir cria e executa uma imagem do Windows Docker usando CodeBuild

Tópicos

- [Execute o exemplo de compilações do Windows Docker](#)

Execute o exemplo de compilações do Windows Docker

Use o procedimento a seguir para executar as compilações do Docker do Windows.

Para executar o Windows, o Docker compila uma amostra

1. Crie os arquivos conforme descrito nas [Arquivos](#) seções [Estrutura de diretório](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

2. Crie uma WINDOWS_EC2 frota.

Se você usar o AWS CLI para criar a frota, a entrada formatada em JSON para o `create-fleet` comando pode ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "fleet-name",
```

```

    "baseCapacity": 1,
    "environmentType": "WINDOWS_EC2",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  }

```

3. Crie um projeto de compilação, execute a compilação e exiba as respectivas informações.

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```

{
  "name": "project-name",
  "source": {
    "type": "S3",
    "location": "bucket-name/DockerImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "WINDOWS_EC2",
    "image": "Windows",
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "fleet": {
      "fleetArn": "fleet-arn"
    }
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}

```

4. Para ver os resultados da compilação, procure no log da compilação a string `Hello, World!`. Para obter mais informações, consulte [Visualizar detalhes de compilação](#).

Estrutura de diretório

Essa amostra pressupõe esta estrutura de diretório.

```

(root directory name)
### buildspec.yml
### Dockerfile

```

Arquivos

A imagem base do sistema operacional usada neste exemplo é `mcr.microsoft.com/windows/servercore:ltsc2022`. A amostra usa estes arquivos.

`buildspec.yml` (em *(root directory name)*)

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld powershell -Command "Write-Host 'Hello World!'"
```

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

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

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

Exemplo de “Publicar imagem do Docker em um repositório de imagens do Amazon ECR” para CodeBuild

Este exemplo produz como saída de compilação de uma imagem do Docker e, depois, a envia a um repositório de imagens do Amazon Elastic Container Registry (Amazon ECR). Você pode adaptar esta amostra para enviar a imagem Docker para um Docker Hub. Para obter mais informações, consulte [Adapte o exemplo “Publicar imagem do Docker no Amazon ECR” para enviar para o Docker Hub](#).

Para saber como compilar uma imagem de Docker usando uma imagem de compilação personalizada do Docker (`docker:dind` no Docker Hub), consulte [Docker no exemplo de imagem personalizada](#).

Esse exemplo foi testado fazendo referência a `golang:1.12`

Esse exemplo usa o novo recurso de compilações do Docker multiestágio, que produz uma imagem de Docker como saída de compilação. Depois, ele envia a imagem do Docker por push a um

repositório de imagens do Amazon ECR. As compilações da imagem de Docker multiestágio ajudam a reduzir o tamanho da imagem de Docker final. Para obter mais informações, consulte [Usar compilações multiestágio com o Docker](#).

Important

A execução dessa amostra pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por AWS CodeBuild AWS recursos e ações relacionados ao Amazon S3 AWS KMS, CloudWatch Logs e Amazon ECR. Para obter mais informações, consulte [CodeBuild preços, preços](#) do [Amazon S3, preços, preços](#) da Amazon e [AWS Key Management Service preços](#) do [Amazon CloudWatch Elastic Container Registry](#).

Tópicos

- [Execute o exemplo “Publicar imagem do Docker no Amazon ECR”](#)
- [Adapte o exemplo “Publicar imagem do Docker no Amazon ECR” para enviar para o Docker Hub](#)

Execute o exemplo “Publicar imagem do Docker no Amazon ECR”

Use o procedimento a seguir para executar o exemplo que publica uma imagem do Docker no Amazon ECR. Para obter mais informações sobre este exemplo, consulte [Exemplo de “Publicar imagem do Docker em um repositório de imagens do Amazon ECR” para CodeBuild](#).

Como executar este exemplo

1. Se você já possui um repositório de imagens no Amazon ECR que deseje usar, avance para a etapa 3. Caso contrário, se você estiver usando um usuário em vez de uma conta AWS raiz ou um usuário administrador para trabalhar com o Amazon ECR, adicione essa declaração (entre **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) ao usuário (ou grupo do IAM ao qual o usuário está associado). Não é recomendável usar uma conta AWS raiz. Essa declaração permite a criação de repositórios Amazon ECR para armazenar imagens do Docker. As elipses (. . .) são usadas para agilizar e para ajudá-lo a encontrar onde adicionar a declaração. Não remova nenhuma declaração e não digite essas elipses nas políticas. Para obter mais informações, consulte [Working with inline policies using the AWS Management Console](#) no Guia do usuário.

```
{
```

```

"Statement": [
  ### BEGIN ADDING STATEMENT HERE ###
  {
    "Action": [
      "ecr:CreateRepository"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  ### END ADDING STATEMENT HERE ###
  ...
],
"Version": "2012-10-17"
}

```

Note

A entidade do IAM que modifica essa política deve ter permissão no IAM para alterar políticas.

2. Crie um repositório de imagens no Amazon ECR. Certifique-se de criar o repositório na mesma AWS região em que você cria seu ambiente de compilação e executa sua compilação. Para obter mais informações, consulte [Creating a repository](#) no Guia do usuário do Amazon ECR. O nome desse repositório deve corresponder ao nome do repositório que você especificará posteriormente neste procedimento, representado pela variável de ambiente IMAGE_REPO_NAME. Certifique-se de que a política de repositório do Amazon ECR conceda acesso push de imagens para sua função CodeBuild de serviço IAM.
3. Adicione essa declaração (entre *### BEGIN ADDING STATEMENT HERE ###* e *### END ADDING STATEMENT HERE ###*) à política que você anexou à sua função AWS CodeBuild de serviço. Essa declaração permite CodeBuild fazer o upload de imagens do Docker para os repositórios do Amazon ECR. As elipses (. . .) são usadas para agilizar e para ajudá-lo a encontrar onde adicionar a declaração. Não remova nenhuma declaração e não digite essas elipses nas políticas.

```

{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:BatchCheckLayerAvailability",

```

```

        "ecr:CompleteLayerUpload",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
### END ADDING STATEMENT HERE ###
...
],
"Version": "2012-10-17"
}

```

Note

A entidade do IAM que modifica essa política deve ter permissão no IAM para alterar políticas.

4. Crie os arquivos conforme descrito nas [Arquivos](#) seções [Estrutura de diretório](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub Para obter mais informações, consulte [Image definitions file reference](#) no Guia do usuário do AWS CodePipeline .

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

5. Crie um projeto de compilação, execute a compilação e exiba suas informações.

Se você usar o console para criar seu projeto:

- a. Para Operating system, selecione Ubuntu.
- b. Em Runtime (Tempo de execução), selecione Standard (Padrão).
- c. Para Imagem, escolha: 5.0aws/codebuild/standard.

d. Adicione as seguintes variáveis de ambiente:

- AWS_DEFAULT_REGION com um valor de *region-ID*
- AWS_ACCOUNT_ID com um valor de *account-ID*
- IMAGE_TAG com o valor Latest
- IMAGE_REPO_NAME com um valor de *Amazon-ECR-repo-name*

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "sample-docker-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      },
      {
        "name": "AWS_ACCOUNT_ID",
        "value": "account-ID"
      },
      {
        "name": "IMAGE_REPO_NAME",
        "value": "Amazon-ECR-repo-name"
      },
      {
        "name": "IMAGE_TAG",
        "value": "latest"
      }
    ]
  }
}
```

```

    ],
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

6. Confirme se a imagem do Docker foi enviada CodeBuild com sucesso para o repositório:

1. Abra o console do Amazon ECR em <https://console.aws.amazon.com/ecr/>.
2. Escolha o nome do repositório. A imagem deve ser listada na coluna Image tag (Tag de imagem)

Estrutura de diretório

Essa amostra pressupõe esta estrutura de diretório.

```

(root directory name)
### buildspec.yml
### Dockerfile

```

Arquivos

Esta amostra usa estes arquivos.

buildspec.yml (em (*root directory name*))

```

version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
    build:
      commands:
        - echo Build started on `date`
        - echo Building the Docker image...
        - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
        - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG

```

```
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
      $IMAGE_REPO_NAME:$IMAGE_TAG
```

Dockerfile (em *(root directory name)*)

```
FROM golang:1.12-alpine AS build
#Install git
RUN apk add --no-cache git
#Get the hello world package from a GitHub repository
RUN go get github.com/golang/example/hello
WORKDIR /go/src/github.com/golang/example/hello
# Build the project and send the output to /bin/HelloWorld
RUN go build -o /bin/HelloWorld

FROM golang:1.12-alpine
#Copy the build's output binary from the previous build container
COPY --from=build /bin/HelloWorld /bin/HelloWorld
ENTRYPOINT ["/bin/HelloWorld"]
```

Note

CodeBuild substitui as imagens personalizadas do ENTRYPOINT Docker.

Adapte o exemplo “Publicar imagem do Docker no Amazon ECR” para enviar para o Docker Hub

Para adaptar o exemplo “Publicar imagem do Docker no Amazon ECR” de modo que a imagem do Docker seja enviada ao Docker Hub e não ao Amazon ECR, edite o código do exemplo. Para obter mais informações sobre o exemplo, consulte [Exemplo de “Publicar imagem do Docker em um repositório de imagens do Amazon ECR” para CodeBuild](#) e [Execute o exemplo “Publicar imagem do Docker no Amazon ECR”](#).

 Note

Se você estiver usando uma versão do Docker anterior à 17.06, remova a opção `--no-include-email`.

1. Substitua estas linhas de código específicas do Amazon ECR no arquivo `buildspec.yml`:

```
...
pre_build:
  commands:
    - echo Logging in to Amazon ECR...
    - aws ecr get-login-password --region $AWS_DEFAULT_REGION |
docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
...
```

Por estas linhas de código específicas no Docker Hub:

```
...
pre_build:
  commands:
    - echo Logging in to Docker Hub...
    # Type the command to log in to your Docker Hub account here.
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
```

```
- docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $IMAGE_REPO_NAME:$IMAGE_TAG
...
```

2. Faça o upload do código editado em um bucket de entrada do S3 ou em um AWS CodeCommit repositório do Bitbucket. GitHub

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

3. Substitua estas linhas de código da entrada formatada JSON para o comando create-project:

```
...
  "environmentVariables": [
    {
      "name": "AWS_DEFAULT_REGION",
      "value": "region-ID"
    },
    {
      "name": "AWS_ACCOUNT_ID",
      "value": "account-ID"
    },
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ]
]
```

```
...
```

Por estas linhas de código:

```
...  
  "environmentVariables": [  
    {  
      "name": "IMAGE_REPO_NAME",  
      "value": "your-Docker-Hub-repo-name"  
    },  
    {  
      "name": "IMAGE_TAG",  
      "value": "latest"  
    }  
  ]  
...
```

4. Crie um ambiente de compilação, execute a compilação e exiba informações relacionadas.
5. Confirme se a imagem do Docker foi enviada AWS CodeBuild com sucesso para o repositório. Faça login no Docker Hub, vá até o repositório e selecione a guia Tags. A tag `latest` deve conter um valor Last Updated bem recente.

Registro privado com AWS Secrets Manager amostra para CodeBuild

Este exemplo mostra como usar uma imagem do Docker armazenada em um registro privado como seu ambiente AWS CodeBuild de execução. As credenciais para o registro privado são armazenadas no AWS Secrets Manager. Qualquer registro privado funciona com CodeBuild. Esse exemplo usa o Docker Hub.

Note

Os segredos são visíveis para ações e não são mascarados quando gravados em um arquivo.

Tópicos

- [Requisitos do exemplo do registro privado](#)
- [Crie um CodeBuild projeto com um registro privado](#)

- [Configurar uma credencial de registro privada para executores auto-hospedados](#)

Requisitos do exemplo do registro privado

Para usar um registro privado com AWS CodeBuild, você deve ter o seguinte:

- Um segredo do Secrets Manager que armazena suas credenciais do Docker Hub. As credenciais são usados para acessar seu repositório privado.

Note

Você será cobrado pelos segredos criados.

- Um repositório ou conta privada.
- Uma política do IAM de função de CodeBuild serviço que concede acesso ao seu segredo do Secrets Manager.

Siga estas etapas para criar esses recursos e, em seguida, criar um projeto de CodeBuild compilação usando as imagens do Docker armazenadas em seu registro privado.

Crie um CodeBuild projeto com um registro privado

1. Para obter informações sobre como criar um repositório privado gratuito, consulte [repositórios no Docker Hub](#). Você também pode executar os comandos a seguir em um terminal para extrair uma imagem, obter seu ID e enviá-la a um novo repositório.

```
docker pull amazonlinux
docker images amazonlinux --format {{.ID}}
docker tag image-id your-username/repository-name:tag
docker login
docker push your-username/repository-name
```

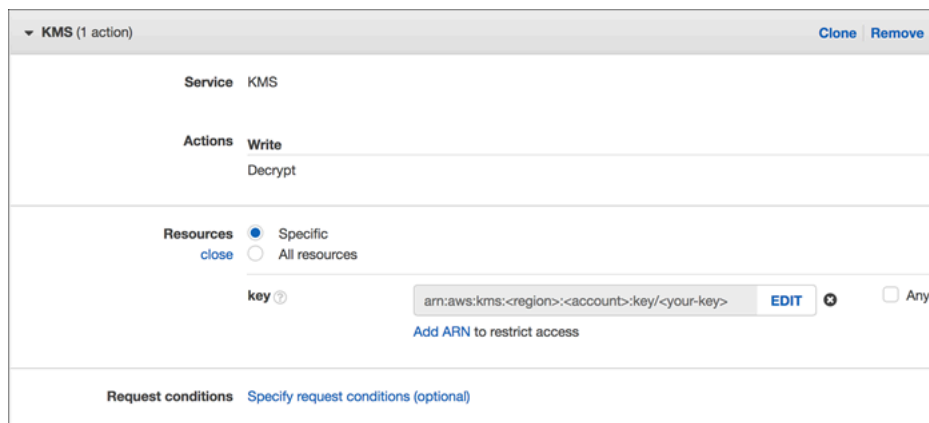
2. Siga as etapas em [Criar um AWS Secrets Manager segredo](#) no Guia do AWS Secrets Manager usuário.
 - a. Na etapa 3, em Escolher tipo de segredo, selecione Outro tipo de segredo.
 - b. Em Chave/valor do segredo, crie um par de chave/valor para o nome de usuário do Docker Hub e um par de chave/valor para a senha do Docker Hub.

- c. Continue seguindo as etapas em [Criar um AWS Secrets Manager segredo](#).
- d. Na etapa 5, na página Configurar alternância automática, desative-a porque as chaves correspondem às credenciais do Docker Hub.
- e. Conclua seguindo as etapas em [Criar um AWS Secrets Manager segredo](#).

Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#)

3. Quando você cria um AWS CodeBuild projeto no console, CodeBuild anexa a permissão necessária para você. Se você usar uma AWS KMS chave diferente de `DefaultEncryptionKey`, deverá adicioná-la à função de serviço. Para obter mais informações, consulte [Modifying a role \(console\)](#) no Guia do usuário do IAM.

Para que o perfil de serviço funcione com o Secrets Manager, ele deverá ter, no mínimo, a permissão `secretsmanager:GetSecretValue`.



4. Para usar o console para criar um projeto com um ambiente armazenado em um registro privado, faça o seguinte enquanto você cria um projeto. Para ter mais informações, consulte [Criar um projeto de compilação \(console\)](#).

Note

Se seu registro privado estiver em sua VPC, ele deverá ter acesso público à Internet. CodeBuild não é possível extrair uma imagem de um endereço IP privado em uma VPC.

- a. Em Imagem de ambiente, selecione Imagem personalizada.
- b. Para Environment type (Tipo de ambiente), selecione Linux ou Windows.
- c. Em Registro de imagens, selecione Outro registro.

- d. Em URL de registro externo, insira o local da imagem e, em Credencial do registro: opcional, insira o ARN ou o nome das credenciais do Secrets Manager.

 Note

Se as suas credenciais não existirem em sua região atual, será necessário usar o ARN. Não será possível usar o nome da credencial se as credenciais existirem em uma região diferente.

Configurar uma credencial de registro privada para executores auto-hospedados

Use as instruções a seguir para configurar uma credencial de registro para um executor auto-hospedado.

 Note

Observe que essas credenciais só serão usadas se as imagens forem substituídas pelas de registros privados.

AWS Management Console

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de construção ou selecione um projeto existente. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Alterar configurações de um projeto de compilação \(console\)](#).
3. Em Ambiente, escolha Configuração adicional.
4. Em Configuração adicional, insira o nome ou ARN do formulário secreto AWS Secrets Manager para a credencial do Registro - opcional.

Registry credential - optional

AWS CLI

1. Se você quiser criar um novo projeto, execute o comando `create-project`.

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

2. Se você quiser atualizar um projeto existente, execute o comando `update-project`.

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

Criar um site estático com saída de compilação hospedada em um bucket do S3

Você pode desabilitar a criptografia de artefatos em uma compilação. Convém fazer isso para que você possa publicar os artefatos em um local configurado para hospedar um site. Não é possível publicar artefatos criptografados. Este exemplo mostra como você pode usar webhooks para acionar uma compilação e publicar seus artefatos em um bucket do S3 configurado como um site.

1. Siga as instruções em [Configurar um site estático](#) para configurar um bucket do S3 para funcionar como um site.
2. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
3. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
4. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição

opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.

5. Em Source (Origem), para Source provider (Provedor de origem), selecione GitHub. Siga as instruções para se conectar (ou reconectar GitHub) e escolha Autorizar.

Em Webhook, selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório). Você só pode marcar essa caixa de seleção caso tenha escolhido Use a repository in my account (Usar um repositório na minha conta).

Source

Add source

Source 1 - Primary

Source provider

GitHub

Repository

☐ Public repository

☒ Repository in my GitHub account

GitHub repository

Disconnect GitHub account

▼ Additional configuration

Git clone depth

Git clone depth - optional

1

Build Status - optional

☐ Report build statuses to source provider when your builds start and finish

Webhook - optional

☒ Rebuild every time a code change is pushed to this repository

Branch filter - optional

Enter a regular expression

6. Em Ambiente:

Para Imagem do ambiente, faça um dos seguintes procedimentos:

- Para usar uma imagem do Docker gerenciada por AWS CodeBuild, escolha Imagem gerenciada e, em seguida, faça seleções em Sistema operacional, Tempo de execução (s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.

- Para usar outra imagem do Docker, selecione Custom image (Imagem personalizada). Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS
- Para usar uma imagem privada do Docker, selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

7. Em Perfil de serviço, faça um dos seguintes procedimentos:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

 Note

Ao usar o console para criar ou atualizar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

8. Em Buildspec, siga um destes procedimentos:

- Escolha Usar um arquivo buildspec para usar o arquivo buildspec.yml no diretório raiz do código-fonte.
- Selecione Inserir comandos de compilação para usar o console para inserir comandos de compilação.

Para obter mais informações, consulte o [Referência de buildspec](#).

9. Em Artefatos, em Tipo, escolha Amazon S3 para armazenar a saída da compilação em um bucket do S3.
10. Em Bucket name (Nome do bucket), escolha o nome do bucket do S3 que você configurou para funcionar como um site na etapa 1.
11. Se você escolheu Insert build commands (Inserir comandos de compilação) em Environment (Ambiente), em Output files (Arquivos de saída), insira os locais dos arquivos da compilação que você quer colocar no bucket de saída. Se você tiver mais de um local, use uma vírgula para separar cada um deles (por exemplo, **appspect.yml**, **target/my-app.jar**). Para obter mais informações, consulte [Artifacts reference-key in the buildspec file](#).
12. Selecione Disable artifacts encryption (Desabilitar a criptografia dos artefatos).
13. Expanda Configuração adicional e escolha as opções conforme apropriado.
14. Selecione Create build project (Criar projeto de compilação). Na página de projeto de compilação, em Build history (Histórico de compilação), escolha Start build (Iniciar compilação) para executar a compilação.
15. (Opcional) Siga as instruções em [Exemplo: Acelere seu site com a Amazon CloudFront](#) no Guia do desenvolvedor do Amazon S3.

Várias fontes de entrada e exemplos de artefatos de saída

Você pode criar um projeto de AWS CodeBuild construção com mais de uma fonte de entrada e mais de um conjunto de artefatos de saída. Este exemplo mostra como configurar um projeto de compilação que:

- Usa várias origens e repositórios de diferentes tipos.
- Publica artefatos de compilação em vários buckets do S3 em uma única compilação.

No exemplo a seguir, você cria um projeto de compilação e o utiliza para executar uma compilação. O exemplo usa o arquivo buildspec do projeto de compilação para mostrar como incorporar mais de uma origem e criar mais de um conjunto de artefatos.

Para saber como criar um pipeline que usa várias entradas de origem CodeBuild para criar vários artefatos de saída, consulte [Amostra de uma CodeBuild integração CodePipeline/com várias fontes de entrada e artefatos de saída](#)

Tópicos

- [Criar um projeto de compilação com várias entradas e saídas](#)
- [Criar um projeto de compilação sem uma origem](#)

Criar um projeto de compilação com várias entradas e saídas

Use o procedimento a seguir para criar um projeto de compilação com várias entradas e saídas.

Para criar um projeto de compilação com várias entradas e saídas

1. Faça upload de suas fontes para um ou mais repositórios S3, CodeCommit, GitHub, GitHub Enterprise Server ou Bitbucket.
2. Escolha qual será a origem principal. Essa é a fonte na qual CodeBuild procura e executa seu arquivo buildspec.
3. Crie um projeto de compilação. Para obter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#).
4. Crie o projeto de compilação, execute a compilação e obtenha informações sobre ela.
5. Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante à seguinte:

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [ss
    {
```

```
    "type": "S3",
    "location": "<output-bucket>",
    "artifactIdentifier": "artifact1"
  },
  {
    "type": "S3",
    "location": "<other-output-bucket>",
    "artifactIdentifier": "artifact2"
  }
],
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Sua origem principal é definida abaixo do atributo `source`. Todas as outras origens são chamadas de secundárias e aparecem abaixo de `secondarySources`. Todas as origens secundárias são instaladas em seu próprio diretório. Esse diretório é armazenado na variável de ambiente interna `CODEBUILD_SRC_DIR_<sourceIdentifier>`. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).

O atributo `secondaryArtifacts` contém uma lista de definições de artefato. Esses artefatos usam o bloco `secondary-artifacts` do arquivo `buildspec` que fica aninhado dentro do bloco `artifacts`.

Os artefatos secundários no arquivo `buildspec` têm a mesma estrutura que os artefatos e são separados pelo seu respectivo identificador de artefato.

Note

Na [API do CodeBuild](#), o `artifactIdentifier` em um artefato secundário é um atributo obrigatório em `CreateProject` e `UpdateProject`. Ele deve ser usado para fazer referência a um artefato secundário.

Com a entrada em formato JSON anterior, o arquivo `buildspec` do projeto deve ter esta aparência:


```
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk11
  build:
    commands:
      - cd $CODEBUILD_SRC_DIR_source1
      - touch file1
      - cd $CODEBUILD_SRC_DIR_source2
      - touch file2

artifacts:
  files:
    - '**.*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR_source1
      files:
        - file1
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - file2
```

Você pode substituir a versão da origem principal usando a API com o atributo `sourceVersion` em `StartBuild`. Para substituir uma ou mais versões de origem secundária, use o atributo `secondarySourceVersionOverride`.

A entrada formatada em JSON para o `start-build` comando no pode ser semelhante a AWS CLI :

```
{
  "projectName": "sample-project",
  "secondarySourcesVersionOverride": [
    {
      "sourceIdentifier": "source1",
      "sourceVersion": "codecommit-branch"
    },
    {
      "sourceIdentifier": "source2",
      "sourceVersion": "github-branch"
    }
  ],
}
```

```
]
}
```

Criar um projeto de compilação sem uma origem

Você pode configurar um CodeBuild projeto escolhendo o tipo de **NO_SOURCE** fonte ao configurar sua fonte. Quando o tipo de origem for **NO_SOURCE**, você não poderá especificar um arquivo buildspec porque o projeto não terá uma origem. Em vez disso, você deverá especificar uma string buildspec formatada em YAML no atributo buildspec da entrada formatada em JSON para o comando `create-project` da CLI. A aparência poderá ser semelhante a esta:

```
{
  "name": "project-name",
  "source": {
    "type": "NO_SOURCE",
    "buildspec": "version: 0.2\n\nphases:\n  build:\n    commands:\n      - command"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).

Versões de tempo de execução na amostra de arquivo buildspec para CodeBuild

Se você usa a imagem padrão do Amazon Linux 2 (AL2) versão 1.0 ou posterior, ou a imagem padrão do Ubuntu versão 2.0 ou posterior, você pode especificar um ou mais tempos de execução na `runtime-versions` seção do seu arquivo buildspec. Os exemplos a seguir mostram como você pode alterar o runtime do projeto, especificar mais de um runtime e especificar um runtime que dependa de outro runtime. Para obter informações sobre tempos de execução com suporte, consulte [Imagens do Docker fornecidas por CodeBuild](#).

Note

Se você usar o Docker no contêiner de compilação, sua compilação deverá ser executada no modo privilegiado. Para ter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#) e [Crie um projeto de construção em AWS CodeBuild](#).

Tópicos

- [Atualizar a versão do runtime no arquivo buildspec](#)
- [Especificar dois tempos de execução](#)

Atualizar a versão do runtime no arquivo buildspec

Modifique o runtime usado pelo projeto para uma nova versão atualizando a seção `runtime-versions` do arquivo `buildspec`. Os exemplos a seguir mostram como especificar as versões 8 e 11 do Java.

- Uma seção `runtime-versions` que especifica a versão 8 de Java:

```
phases:
  install:
    runtime-versions:
      java: corretto8
```

- Uma seção `runtime-versions` que especifica a versão 11 de Java:

```
phases:
  install:
    runtime-versions:
      java: corretto11
```

Os exemplos a seguir mostram como especificar diferentes versões do Python usando a imagem padrão 5.0 do Ubuntu ou a imagem padrão 3.0 do Amazon Linux 2:

- Uma seção `runtime-versions` que especifica o Python versão 3.7:

```
phases:
  install:
```

```
runtime-versions:
  python: 3.7
```

- Uma seção `runtime-versions` que especifica o Python versão 3.8:

```
phases:
  install:
    runtime-versions:
      python: 3.8
```

Este exemplo demonstra um projeto que começa com o tempo de execução Java versão 8 e, depois, é atualizado para o tempo de execução Java versão 10.

1. Faça o download e instale o Maven. Para obter mais informações, consulte [Downloading Apache Maven](#) e [Installing Apache Maven](#) no site da Apache Maven.
2. Alterne para um diretório vazio em seu computador ou instância local e execute este comando Maven.

```
mvn archetype:generate "-DgroupId=com.mycompany.app" "-DartifactId=ROOT" "-DarchetypeArtifactId=maven-archetype-webapp" "-DinteractiveMode=false"
```

Se tiver êxito, a estrutura de diretório e os arquivos serão criados.

```
.
### ROOT
  ### pom.xml
  ### src
    ### main
      ### resources
      ### webapp
        ### WEB-INF
        #   ### web.xml
        ### index.jsp
```

3. Crie um arquivo denominado `buildspec.yml` com o seguinte conteúdo: Armazene o arquivo no diretório *(root directory name)/my-web-app*.

```
version: 0.2

phases:
```

```
install:
  runtime-versions:
    java: corretto8
build:
  commands:
    - java -version
    - mvn package
artifacts:
  files:
    - '**/*'
base-directory: 'target/my-web-app'
```

No arquivo buildspec:

- A seção `runtime-versions` especifica que o projeto usa o tempo de execução Java versão 8.
- O comando `- java -version` exibe a versão de Java usada pelo seu projeto ao ser compilado.

A estrutura de arquivo deve se parecer com esta.

```
(root directory name)
### my-web-app
  ### src
    #   ### main
    #   ### resources
    #   ### webapp
    #       ### WEB-INF
    #           ### web.xml
    #               ### index.jsp
  ### buildspec.yml
  ### pom.xml
```

4. Faça upload do conteúdo do `my-web-app` diretório em um bucket de entrada do S3 ou em um repositório CodeCommit GitHub, ou Bitbucket.

⚠ Important

Não faça upload de *(root directory name)* nem *(root directory name)/my-web-app*, apenas dos diretórios e arquivos em *(root directory name)/my-web-app*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha a estrutura de diretório e os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* nem *(root directory name)/my-web-app* ao arquivo ZIP, apenas aos diretórios e arquivos em *(root directory name)/my-web-app*.

5. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
6. Crie um projeto de compilação. Para ter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#). Deixe todas as configurações em seus valores padrão, exceto estas configurações.
 - Em Ambiente:
 - Em Imagem do ambiente, escolha Imagem gerenciada.
 - Em Sistema operacional, escolha Amazon Linux 2.
 - Em Runtime(s), selecione Padrão.
 - Para Imagem, escolha aws/codebuild/amazonlinux-x86_64-standard:4.0.
7. Selecione Iniciar compilação.
8. Aceite os padrões em Configuração de compilação e escolha Iniciar compilação.
9. Assim que a compilação for concluída, veja a saída da compilação na guia Logs de compilação. Você deve ver uma saída semelhante a:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto8' based on manual selections...
[Container] Date Time Running command echo "Installing Java version 8 ..."
Installing Java version 8 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_8_HOME"
```

```
[Container] Date Time Running command export JRE_HOME="$JRE_8_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_8_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_8_HOME"/bin/*
"$JRE_8_HOME"/bin/*;
```

10. Atualize a seção runtime-versions para Java versão 11:

```
install:
  runtime-versions:
    java: corretto11
```

11. Depois de salvar a alteração, execute a compilação novamente e veja a saída da compilação. A versão instalada de Java deverá ser 11. Você deve ver uma saída semelhante a:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual
selections...
Installing Java version 11 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*
"$JRE_11_HOME"/bin/*;
```

Especificar dois tempos de execução

Você pode especificar mais de um tempo de execução no mesmo projeto de CodeBuild compilação. Este exemplo de projeto usa dois arquivos de origem: um que usa o tempo de execução Go e um que usa o tempo de execução Node.js.

1. Crie um diretório chamado my-source.

2. No diretório `my-source`, crie um diretório chamado `golang-app`.
3. Crie um arquivo denominado `hello.go` com o seguinte conteúdo: Armazene o arquivo no diretório `golang-app`.

```
package main
import "fmt"

func main() {
    fmt.Println("hello world from golang")
    fmt.Println("1+1 =", 1+1)
    fmt.Println("7.0/3.0 =", 7.0/3.0)
    fmt.Println(true && false)
    fmt.Println(true || false)
    fmt.Println(!true)
    fmt.Println("good bye from golang")
}
```

4. No diretório `my-source`, crie um diretório chamado `nodejs-app`. Ele deve estar no mesmo nível que o diretório `golang-app`.
5. Crie um arquivo denominado `index.js` com o seguinte conteúdo: Armazene o arquivo no diretório `nodejs-app`.

```
console.log("hello world from nodejs");
console.log("1+1 =" + (1+1));
console.log("7.0/3.0 =" + 7.0/3.0);
console.log(true && false);
console.log(true || false);
console.log(!true);
console.log("good bye from nodejs");
```

6. Crie um arquivo denominado `package.json` com o seguinte conteúdo: Armazene o arquivo no diretório `nodejs-app`.

```
{
  "name": "mycompany-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"run some tests here\""
  },
}
```



```
"author": "",  
"license": "ISC"  
}
```

7. Crie um arquivo denominado `buildspec.yml` com o seguinte conteúdo: Armazene o arquivo no diretório `my-source`, no mesmo nível que os diretórios `nodejs-app` e `golang-app`. A seção `runtime-versions` especifica os runtimes do Node.js versão 12 e do Go versão 1.13.

```
version: 0.2  
  
phases:  
  install:  
    runtime-versions:  
      golang: 1.13  
      nodejs: 12  
    build:  
      commands:  
        - echo Building the Go code...  
        - cd $CODEBUILD_SRC_DIR/golang-app  
        - go build hello.go  
        - echo Building the Node code...  
        - cd $CODEBUILD_SRC_DIR/nodejs-app  
        - npm run test  
  artifacts:  
    secondary-artifacts:  
      golang_artifacts:  
        base-directory: golang-app  
        files:  
          - hello  
      nodejs_artifacts:  
        base-directory: nodejs-app  
        files:  
          - index.js  
          - package.json
```

8. A estrutura de arquivo deve se parecer com esta.

```
my-source  
### golang-app  
#   ### hello.go  
### nodejs.app  
#   ### index.js  
#   ### package.json
```

```
### buildspec.yml
```

9. Faça upload do conteúdo do my-source diretório em um bucket de entrada do S3 ou em um repositório CodeCommit GitHub, ou Bitbucket.

 Important

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha a estrutura de diretório e os arquivos e envie-o para o bucket de entrada. Não adicione my-source ao arquivo ZIP, apenas aos diretórios e arquivos em my-source.

10. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
11. Crie um projeto de compilação. Para ter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#). Deixe todas as configurações em seus valores padrão, exceto estas configurações.
 - Em Ambiente:
 - Em Imagem do ambiente, escolha Imagem gerenciada.
 - Em Sistema operacional, escolha Amazon Linux 2.
 - Em Runtime(s), selecione Padrão.
 - Para Imagem, escolha aws/codebuild/amazonlinux-x86_64-standard:4.0.
12. Selecione Create build project (Criar projeto de compilação).
13. Selecione Iniciar compilação.
14. Aceite os padrões em Configuração de compilação e escolha Iniciar compilação.
15. Assim que a compilação for concluída, veja a saída da compilação na guia Logs de compilação. Você deve ver saída semelhante ao seguinte: Ela mostra a saída dos tempos de execução Go e Node.js. Ela também mostra a saída dos aplicativos Go e Node.js.

```
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'golang' runtime version '1.13' based on manual
selections...
[Container] Date Time Selecting 'nodejs' runtime version '12' based on manual
selections...
[Container] Date Time Running command echo "Installing Go version 1.13 ..."
Installing Go version 1.13 ...

[Container] Date Time Running command echo "Installing Node.js version 12 ..."
Installing Node.js version 12 ...
```

```
[Container] Date Time Running command n $NODE_12_VERSION
    installed : v12.20.1 (with npm 6.14.10)

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time   INSTALL: 0 commands
[Container] Date Time   BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code:   Message:
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
[Container] Date Time Phase context status code:   Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code:   Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/golang-app

[Container] Date Time Running command go build hello.go

[Container] Date Time Running command echo Building the Node code...
Building the Node code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/nodejs-app

[Container] Date Time Running command npm run test

> mycompany-app@1.0.0 test /codebuild/output/src924084119/src/nodejs-app
> echo "run some tests here"

run some tests here
```

Amostra da versão de origem com AWS CodeBuild

Esse exemplo demonstra como especificar uma versão de sua origem usando um formato diferente de um ID de confirmação (também conhecido como um SHA de confirmação). Você pode especificar a versão da sua fonte das seguintes maneiras:

- Para um provedor de origem do Amazon S3, use o ID da versão do objeto que representa o arquivo ZIP de entrada da compilação.
- Para CodeCommit o Bitbucket e o GitHub Enterprise Server, use um dos seguintes: GitHub
 - Solicitação pull como uma referência de solicitação pull (por exemplo, `refs/pull/1/head`).
 - Ramificação como um nome de ramificação.
 - ID de confirmação
 - Tag.
- Referência e um ID de confirmação. A referência pode ser uma das seguintes:
 - Uma tag (por exemplo, `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Uma ramificação (por exemplo, `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Uma solicitação pull (por exemplo, `refs/pull/1/head^{full-commit-SHA}`).
- Para GitLab e GitLab autogerenciado, use um dos seguintes:
 - Ramificação como um nome de ramificação.
 - ID de confirmação
 - Tag.

 Note

Você pode especificar a versão de uma fonte de pull request somente se seu repositório for GitHub ou GitHub Enterprise Server.

Se você usar uma referência e um ID de confirmação para especificar uma versão, a fase `DOWNLOAD_SOURCE` da sua compilação será mais rápida do que se você fornecer somente a versão. Isso ocorre porque quando você adiciona uma referência, CodeBuild não é necessário baixar o repositório inteiro para encontrar o commit.

- Você pode especificar uma versão de origem com apenas um ID de confirmação, como `12345678901234567890123467890123456789`. Se você fizer isso, CodeBuild deverá baixar o repositório inteiro para encontrar a versão.
- Você pode especificar uma versão de origem com uma referência e um ID de confirmação neste formato: `refs/heads/branchname^{full-commit-SHA}` (por exemplo, `refs/heads/main^{12345678901234567890123467890123456789}`). Se você fizer isso, CodeBuild baixará somente a ramificação especificada para encontrar a versão.

 Note

Para acelerar a `DOWNLOAD_SOURCE` fase da sua compilação, você também pode definir a profundidade do clone do Git para um número baixo. CodeBuild baixa menos versões do seu repositório.

Tópicos

- [Especifique uma versão GitHub do repositório com um ID de confirmação](#)
- [Especifique uma versão do GitHub repositório com uma referência e um ID de confirmação](#)

Especifique uma versão GitHub do repositório com um ID de confirmação

Você pode especificar uma versão de origem com apenas um ID de confirmação, como `12345678901234567890123467890123456789`. Se você fizer isso, CodeBuild deverá baixar o repositório inteiro para encontrar a versão.

Para especificar uma versão GitHub do repositório com um ID de confirmação

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#). Deixe todas as configurações em seus valores padrão, exceto estas configurações:
 - Em Origem:
 - Em Source provider, escolha GitHub. Se você não estiver conectado GitHub, siga as instruções para se conectar.
 - Em Repository (Repositório), escolha Public repository (Repositório público).
 - Em URL do repositório, insira **`https://github.com/aws/aws-sdk-ruby.git`**.
 - Em Ambiente:
 - Em Imagem do ambiente, escolha Imagem gerenciada.
 - Em Sistema operacional, escolha Amazon Linux 2.
 - Em Runtime(s), selecione Padrão.
 - Para Imagem, escolha `aws/codebuild/amazonlinux-x86_64-standard:4.0`.

3. Em Build specifications (Especificações de compilação), escolha Insert build commands (Inserir comandos de compilação) e selecione Switch to editor (Alternar para editor).
4. Em Build commands (Comandos de compilação), substitua o texto de espaço reservado pelo seguinte:

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  build:
    commands:
      - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

A seção `runtime-versions` é necessária quando você usa a imagem padrão do Ubuntu 2.0. Aqui, o tempo de execução da versão 2.6 do Ruby é especificado, mas você pode usar qualquer tempo de execução. O comando `echo` exibe a versão do código-fonte armazenada na variável de ambiente `CODEBUILD_RESOLVED_SOURCE_VERSION`.

5. Aceite os padrões em Configuração de compilação e escolha Iniciar compilação.
6. Para Source version (Versão de origem), insira **046e8b67481d53bdc86c3f6affdd5d1afae6d369**. Esse é o SHA de uma confirmação no repositório <https://github.com/aws/aws-sdk-ruby.git>.
7. Selecione Iniciar compilação.
8. Quando a compilação for concluída, você provavelmente verá o seguinte:
 - Na guia Logs de compilação, qual versão do projeto de origem foi usada. Aqui está um exemplo.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Na guia Variáveis de ambiente, a Versão de origem resolvida corresponde ao ID de confirmação usado para criar a compilação.
- Na guia Phase details (Detalhes da fase), a duração da fase `DOWNLOAD_SOURCE`.

Especifique uma versão do GitHub repositório com uma referência e um ID de confirmação

Você pode especificar uma versão de origem com uma referência e um ID de confirmação neste formato: ***refs/heads/branchname^{full-commit-SHA}*** (por exemplo, ***refs/heads/main^{12345678901234567890123467890123456789}***). Se você fizer isso, CodeBuild baixará somente a ramificação especificada para encontrar a versão.

Para especificar uma versão do GitHub repositório com uma referência e um ID de confirmação.

1. Siga as etapas em [Especifique uma versão GitHub do repositório com um ID de confirmação](#).
2. No painel de navegação esquerdo, escolha Build projects (Projetos de compilação) e depois escolha o projeto criado anteriormente.
3. Selecione Iniciar compilação.
4. Na Source version (Versão de origem), insira ***refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}***. Trata-se do mesmo ID de confirmação e de uma referência para uma ramificação no formato ***refs/heads/branchname^{full-commit-SHA}***.
5. Selecione Iniciar compilação.
6. Quando a compilação for concluída, você provavelmente verá o seguinte:
 - Na guia Logs de compilação, qual versão do projeto de origem foi usada. Aqui está um exemplo.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Na guia Variáveis de ambiente, a Versão de origem resolvida corresponde ao ID de confirmação usado para criar a compilação.
- Na guia Phase details (Detalhes da fase), a duração da fase DOWNLOAD_SOURCE deve ser menor que a duração em que você usou apenas o ID de confirmação para especificar a versão de sua origem.

Amostras de repositórios de origem de terceiros para CodeBuild

Esta seção descreve exemplos de integrações entre repositórios de origem de terceiros e CodeBuild

Amostra	Descrição
BitBucket amostra de pull request e filtro de webhook — consulte Execute a amostra “Solicitação de pull e filtro de webhook do Bitbucket” para CodeBuild	Este exemplo mostra como criar uma solicitação usando um repositório do Bitbucket. Ele também mostra como usar um Bitbucket webhook para acionar CodeBuild para criar uma compilação de um projeto.
GitHub Exemplo de servidor corporativo — consulte Execute a amostra do GitHub Enterprise Server para CodeBuild	Este exemplo mostra como configurar seus CodeBuild projetos quando o repositório do GitHub Enterprise Server tem um certificado instalado. Também mostra como habilitar webhooks para que CodeBuild reconstrua o código-fonte toda vez que uma alteração de código for enviada ao seu GitHub repositório do Enterprise Server.
GitHub amostra de pull request e filtro de webhook — consulte Execute a solicitação GitHub pull e a amostra do filtro de webhook para CodeBuild	Este exemplo mostra como criar uma pull request usando um repositório do GitHub Enterprise Server. Também mostra como habilitar webhooks para que CodeBuild reconstrua o código-fonte toda vez que uma alteração de código for enviada ao seu GitHub repositório do Enterprise Server.

Execute a amostra “Solicitação de pull e filtro de webhook do Bitbucket” para CodeBuild

AWS CodeBuild suporta webhooks quando o repositório de origem é o Bitbucket. Isso significa que, para um projeto de CodeBuild compilação que tem seu código-fonte armazenado em um repositório do Bitbucket, os webhooks podem ser usados para reconstruir o código-fonte toda vez que uma

alteração de código é enviada para o repositório. Para obter mais informações, consulte [Filtrar eventos de webhook do Bitbucket](#).

Este exemplo mostra como criar uma solicitação usando um repositório do Bitbucket. Também mostra como usar um webhook do Bitbucket para acionar CodeBuild a criação de uma compilação de um projeto.

Note

Ao usar webhooks, é possível que um usuário acione uma compilação inesperada. Para atenuar esse risco, consulte [Práticas recomendadas para usar webhooks](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um projeto de compilação com o Bitbucket e habilitar os webhooks](#)
- [Etapa 2: acionar uma compilação com um webhook do Bitbucket](#)

Pré-requisitos

Para executar essa amostra, você deve conectar seu AWS CodeBuild projeto à sua conta do Bitbucket.

Note

CodeBuild atualizou suas permissões com o Bitbucket. Se você conectou seu projeto anteriormente ao Bitbucket e agora recebe um erro de conexão do Bitbucket, você deve se reconectar para conceder CodeBuild permissão para gerenciar seus webhooks.

Etapa 1: criar um projeto de compilação com o Bitbucket e habilitar os webhooks

As etapas a seguir descrevem como criar um AWS CodeBuild projeto com o Bitbucket como repositório de origem e habilitar webhooks.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Selecione Create build project (Criar projeto de compilação).
4. Em Configuração de projetos:

Nome do projeto

Insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.

5. Em Origem:

Provedor de origem

Selecione Bitbucket. Siga as instruções para se conectar (ou se reconectar) ao Bitbucket e depois escolha Autorizar.

Repositório

Escolha Repositório em minha conta Bitbucket.

Se você ainda não se conectou à conta do Bitbucket, insira o nome de usuário e senha do aplicativo do Bitbucket e selecione Salvar credenciais do Bitbucket.

Repositório do Bitbucket

Insira o URL do repositório do Bitbucket.

6. Em Eventos de webhook da origem principal, selecione o seguinte.

Note

A seção Eventos de webhook da origem principal só estará visível se você escolheu Repositório em minha conta Bitbucket na etapa anterior.

1. Selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório) ao criar seu projeto.
2. Em Event type (Tipo de evento), escolha um ou mais eventos.

3. Para filtrar quando um evento aciona uma compilação, em Start a build under these conditions (Iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
4. Para filtrar quando um evento não é acionado, em Don't start a build under these conditions (Não iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
5. Escolha Adicionar grupo de filtros para adicionar outro grupo de filtros, se necessário.

Para obter mais informações sobre os tipos e os filtros de eventos de webhook do Bitbucket, consulte [Filtrar eventos de webhook do Bitbucket](#).

7. Em Ambiente:

Imagem do ambiente

Escolha uma das seguintes opções:

Para usar uma imagem do Docker gerenciada por: AWS CodeBuild

Selecione Imagem gerenciada e, depois, faça as seleções em Sistema operacional, Runtime(s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.

Para usar outra imagem do Docker:

Selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS

Para usar uma imagem privada do Docker:

Selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é AWS Secrets Manager?](#) no Guia do AWS Secrets Manager usuário.

Perfil de serviço

Escolha uma das seguintes opções:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

 Note

Ao usar o console para criar ou atualizar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

8. Em Buildspec, siga um destes procedimentos:

- Escolha Usar um arquivo buildspec para usar o arquivo buildspec.yml no diretório raiz do código-fonte.
- Selecione Inserir comandos de compilação para usar o console para inserir comandos de compilação.

Para obter mais informações, consulte [Referência de buildspec](#).

9. Em Artefatos:

Tipo

Escolha uma das seguintes opções:

- Se você não quiser criar artefatos de saída de compilação, selecione Nenhum artefato.
- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:
 - Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. Por padrão, o nome do artefato é o nome do projeto. Se você quiser usar um nome diferente, insira-o na caixa de nome do artefato. Se você quiser gerar um arquivo ZIP, inclua a extensão zip.
 - Para Bucket name, selecione o nome do bucket de saída.

- Se você tiver escolhido Inserir comandos de compilação anteriormente neste procedimento, em Arquivos de saída, insira os locais dos arquivos da compilação que deseja incluir na pasta ou no arquivo ZIP de saída da compilação. Para vários locais, separe-os com uma vírgula (por exemplo, `appspec.yml`, `target/my-app.jar`). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).

Configuração adicional

Expanda Configuração adicional e defina as opções conforme apropriado.

10. Selecione Create build project (Criar projeto de compilação). Na página Revisão, escolha Iniciar compilação para executar a compilação.

Etapa 2: acionar uma compilação com um webhook do Bitbucket

Para um projeto que usa webhooks do Bitbucket, AWS CodeBuild cria uma compilação quando o repositório do Bitbucket detecta uma alteração no seu código-fonte.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Build projects (Projetos de compilação) e, em seguida, escolha um projeto associado a um repositório de Bitbucket com webhooks. Para obter informações sobre como criar um projeto de webhook do Bitbucket, consulte [the section called “Etapa 1: criar um projeto de compilação com o Bitbucket e habilitar os webhooks”](#).
3. Faça algumas alterações no código no repositório do Bitbucket do seu projeto.
4. Criar uma solicitação em seu repositório do Bitbucket. Para obter mais informações, consulte [Making a pull request](#).
5. Na página Webhooks do Bitbucket, escolha View request (Visualizar solicitação) para ver uma lista de eventos recentes.
6. Escolha Exibir detalhes para ver detalhes sobre a resposta retornada por CodeBuild. A aparência pode ser semelhante a esta:

```
"response": "Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
"statusCode": 200
```

7. Navegue até a página de solicitação de Bitbucket para ver o status da compilação.

Execute a amostra do GitHub Enterprise Server para CodeBuild

AWS CodeBuild suporta o GitHub Enterprise Server como um repositório de origem. Este exemplo mostra como configurar seus CodeBuild projetos quando o repositório do GitHub Enterprise Server tem um certificado instalado. Também mostra como habilitar webhooks para que CodeBuild reconstrua o código-fonte toda vez que uma alteração de código for enviada ao seu GitHub repositório do Enterprise Server.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um projeto de compilação com o GitHub Enterprise Server e habilitar webhooks](#)

Pré-requisitos

1. Gere um token de acesso pessoal para seu CodeBuild projeto. Recomendamos que você crie um usuário GitHub corporativo e gere um token de acesso pessoal para esse usuário. Copie-o para sua prancheta para que ele possa ser usado ao criar seu CodeBuild projeto. Para obter mais informações, consulte [Criação de um token de acesso pessoal para a linha de comando](#) no site da GitHub Ajuda.

Ao criar o token de acesso pessoal, inclua o escopo do repositório na definição.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

☒	repo	Full control of private repositories
☒	repo:status	Access commit status
☒	repo_deployment	Access deployment status
☒	public_repo	Access public repositories

2. Baixe seu certificado do GitHub Enterprise Server. CodeBuild usa o certificado para criar uma conexão SSL confiável com o repositório.

Clientes do Linux/macOS:

Na janela de terminal do , execute o seguinte comando:

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```

Substitua os espaços reservados no comando pelos seguintes valores:

HOST. O endereço IP do seu repositório do GitHub Enterprise Server.

PORTNUMBER. O número da porta que você está usando para se conectar (por exemplo, 443).

folder. A pasta em que você baixou seu certificado.

filename. O nome do arquivo do seu certificado.

 Important

Salve o certificado como um arquivo .pem.

Clientes no Windows:

Use seu navegador para baixar seu certificado do GitHub Enterprise Server. Para ver os detalhes do certificado do site, escolha o ícone de cadeado. Para obter informações sobre como exportar o certificado, consulte a documentação do navegador.

 Important

Salve o certificado como um arquivo .pem.

3. Faça upload do arquivo do certificado para um bucket do S3. Para obter informações sobre como criar um bucket do S3, consulte [Como criar um bucket do S3?](#) Para obter informações sobre como fazer upload de objetos para um bucket do S3, consulte [Como fazer upload de arquivos e pastas para um bucket?](#)

 Note

Esse bucket deve estar na mesma AWS região das suas compilações. Por exemplo, se você CodeBuild instruir a execução de uma compilação na região Leste dos EUA (Ohio), o bucket deverá estar na região Leste dos EUA (Ohio).

Etapa 1: criar um projeto de compilação com o GitHub Enterprise Server e habilitar webhooks

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.
4. Em Origem, em Provedor de origem, escolha Servidor GitHub corporativo.
 - Escolha Gerenciar credenciais da conta e, em seguida, escolha Token de acesso pessoal. Em Service, escolha Secrets Manager (recomendado) e configure seu segredo. Em seguida, em Token de acesso pessoal GitHub corporativo, insira seu token de acesso pessoal e escolha Salvar.
 - Em Repository URL (URL do repositório), insira o caminho para o seu repositório, incluindo o nome do repositório.
 - Expanda Configuração Adicional.
 - Selecione Rebuild every time a code change is pushed to this repository para recompilar toda vez que uma alteração de código for enviada para esse repositório.
 - Selecione Habilitar SSL inseguro para ignorar os avisos de SSL enquanto você se conecta ao seu repositório de projetos do GitHub Enterprise Server.

Note

Recomendamos usar Enable insecure SSL somente para teste. Ele não deve ser usado em um ambiente de produção.

Source

Add source

Source 1 - Primary

Source provider

GitHub Enterprise▼

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ Additional configuration
Git clone depth, Insecure SSL

Git clone depth - *optional*

1▼

Webhook - *optional*

☒ Rebuild every time a code change is pushed to this repository

Branch filter - *optional*

Enter a regular expression

Insecure SSL - *optional*

Enable this flag to ignore SSL warnings while connecting to project source.

☐ Enable insecure SSL

5. Em Ambiente:

Para Imagem do ambiente, faça um dos seguintes procedimentos:

- Para usar uma imagem do Docker gerenciada por AWS CodeBuild, escolha Imagem gerenciada e, em seguida, faça seleções em Sistema operacional, Tempo de execução (s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.

- Para usar outra imagem do Docker, selecione Custom image (Imagem personalizada). Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS
 - Para usar uma imagem privada do Docker, selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .
6. Em Perfil de serviço, faça um dos seguintes procedimentos:
- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
 - Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

 Note

Ao usar o console para criar ou atualizar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

7. Expanda Configuração Adicional.

Se você quiser CodeBuild trabalhar com sua VPC:

- Para VPC, escolha a ID da VPC que usa. CodeBuild
- Para sub-redes VPC, escolha as sub-redes que incluem os recursos que usa. CodeBuild
- Para grupos de segurança da VPC, escolha os grupos de segurança CodeBuild usados para permitir o acesso aos recursos no. VPCs

Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

8. Em Buildspec, siga um destes procedimentos:

- Escolha Usar um arquivo buildspec para usar o arquivo buildspec.yml no diretório raiz do código-fonte.
- Selecione Inserir comandos de compilação para usar o console para inserir comandos de compilação.

Para obter mais informações, consulte o [Referência de buildspec](#).

9. Em Artefatos, para Tipo, faça um dos seguintes procedimentos:

- Se você não quiser criar artefatos de saída de compilação, selecione Nenhum artefato.
- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:
 - Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. Por padrão, o nome do artefato é o nome do projeto. Se você quiser usar um nome diferente, insira-o na caixa de nome do artefato. Se você quiser gerar um arquivo ZIP, inclua a extensão zip.
 - Para Bucket name, selecione o nome do bucket de saída.
 - Se você tiver escolhido Inserir comandos de compilação anteriormente neste procedimento, em Arquivos de saída, insira os locais dos arquivos da compilação que deseja incluir na pasta ou no arquivo ZIP de saída da compilação. Para vários locais, separe-os com uma vírgula (por exemplo, `appspec.yml`, `target/my-app.jar`). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).

10. Em Cache type (Tipo de cache), selecione uma das seguintes opções:

- Se não quiser usar um cache, escolha No cache.
- Se você quiser um cache do Amazon S3, selecione Amazon S3 e faça o seguinte:
 - Em Bucket, escolha o nome do bucket do S3 onde o cache está armazenado.
 - (Opcional) Em Prefixo do caminho do cache, insira um prefixo de caminho do Amazon S3. O valor Cache path prefix (Prefixo do caminho do cache) é semelhante ao nome de um diretório. Ele permite que você armazene o cache em um bucket no mesmo diretório.

 Important

Não acrescente uma barra (/) ao final do prefixo do caminho.

- Se você quiser usar um cache local, selecione Local e depois selecione um ou mais modos de cache local.

 Note

O modo de cache de camada do Docker está disponível apenas para o Linux. Se optar por esse modo, o projeto deverá ser executado no modo privilegiado.

O uso do cache economiza um tempo de compilação considerável porque as partes reutilizáveis do ambiente de compilação são armazenadas no cache e usadas em diferentes compilações. Para obter informações sobre como especificar um cache no arquivo de especificação de compilação, consulte [Sintaxe de buildspec](#). Para obter mais informações sobre armazenamento em cache, consulte [Compilações em cache para melhorar o desempenho](#).

11. Selecione Create build project (Criar projeto de compilação). Na página de projeto da compilação, escolha Start build (Iniciar compilação).

Execute a solicitação GitHub pull e a amostra do filtro de webhook para CodeBuild

AWS CodeBuild suporta webhooks quando o repositório de origem está. GitHub Isso significa que, para um projeto de CodeBuild compilação que tem seu código-fonte armazenado em um GitHub repositório, os webhooks podem ser usados para reconstruir o código-fonte toda vez que uma alteração de código é enviada para o repositório. Para obter CodeBuild amostras, consulte [AWS CodeBuild Amostras](#).

 Note

Ao usar webhooks, é possível que um usuário acione uma compilação inesperada. Para atenuar esse risco, consulte [Práticas recomendadas para usar webhooks](#).

Tópicos

- [Etapa 1: criar um projeto de compilação com GitHub webhooks e ativá-los](#)
- [Etapa 2: verificar se os webhooks estão habilitados](#)

Etapa 1: criar um projeto de compilação com GitHub webhooks e ativá-los

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Selecione Create build project (Criar projeto de compilação).
4. Em Configuração de projetos:

Nome do projeto

Insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.

5. Em Origem:

Provedor de origem

Selecione GitHub. Siga as instruções para se conectar (ou reconectar) GitHub e escolha Autorizar.

Repositório

Escolha Repositório em minha GitHub conta.

GitHub repositório

Insira a URL do seu GitHub repositório.

6. Em Eventos de webhook da origem principal, selecione o seguinte.

Note

A seção Eventos de webhook de origem primária só estará visível se você escolher Repositório em minha GitHub conta na etapa anterior.

1. Selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório) ao criar seu projeto.
2. Em Event type (Tipo de evento), escolha um ou mais eventos.
3. Para filtrar quando um evento aciona uma compilação, em Start a build under these conditions (Iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
4. Para filtrar quando um evento não é acionado, em Don't start a build under these conditions (Não iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
5. Escolha Adicionar grupo de filtros para adicionar outro grupo de filtros, se necessário.

Para obter mais informações sobre filtros e tipos de eventos de GitHub webhook, consulte [GitHub eventos de webhook](#).

7. Em Ambiente:

Imagem do ambiente

Escolha uma das seguintes opções:

Para usar uma imagem do Docker gerenciada por: AWS CodeBuild

Selecione Imagem gerenciada e, depois, faça as seleções em Sistema operacional, Runtime(s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.

Para usar outra imagem do Docker:

Selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher Amazon ECR, use Repositório do Amazon ECR e Imagem do Amazon ECR para selecionar a imagem do Docker na conta da AWS .

Para usar uma imagem privada do Docker:

Selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As

credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é AWS Secrets Manager?](#) no Guia do AWS Secrets Manager usuário.

Perfil de serviço

Escolha uma das seguintes opções:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

Note

Ao usar o console para criar ou atualizar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

8. Em Buildspec, siga um destes procedimentos:

- Escolha Usar um arquivo buildspec para usar o arquivo buildspec.yml no diretório raiz do código-fonte.
- Selecione Inserir comandos de compilação para usar o console para inserir comandos de compilação.

Para obter mais informações, consulte [Referência de buildspec](#).

9. Em Artefatos:

Tipo

Escolha uma das seguintes opções:

- Se você não quiser criar artefatos de saída de compilação, selecione Nenhum artefato.
- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:

- Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. Por padrão, o nome do artefato é o nome do projeto. Se você quiser usar um nome diferente, insira-o na caixa de nome do artefato. Se você quiser gerar um arquivo ZIP, inclua a extensão zip.
- Para Bucket name, selecione o nome do bucket de saída.
- Se você tiver escolhido Inserir comandos de compilação anteriormente neste procedimento, em Arquivos de saída, insira os locais dos arquivos da compilação que deseja incluir na pasta ou no arquivo ZIP de saída da compilação. Para vários locais, separe-os com uma vírgula (por exemplo, `appspec.yml`, `target/my-app.jar`). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).

Configuração adicional

Expanda Configuração adicional e defina as opções conforme apropriado.

10. Selecione Create build project (Criar projeto de compilação). Na página Revisão, escolha Iniciar compilação para executar a compilação.

Etapa 2: verificar se os webhooks estão habilitados

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Execute um destes procedimentos:
 - Selecione o link do projeto de compilação com webhooks que você deseja verificar e selecione Detalhes da compilação.
 - Selecione o botão ao lado do projeto de compilação com webhooks que você deseja verificar, selecione Visualizar detalhes e, depois, Detalhes da compilação.
4. Em Eventos de webhook da origem principal, escolha o link URL do Webhook.
5. No seu GitHub repositório, na página Configurações, em Webhooks, verifique se Pull Requests e Pushes estão selecionados.
6. Nas configurações GitHub do seu perfil, em Configurações pessoais, Aplicativos, OAuthAplicativos autorizados, você deve ver que seu aplicativo foi autorizado a acessar a AWS região selecionada.

Tutorial: Assinatura de código da Apple com Fastlane ao CodeBuild usar o S3 para armazenamento de certificados

[fastlane](#) é uma ferramenta popular de automação de código aberto para automatizar implantações e lançamentos beta para seus aplicativos iOS e Android. Ele lida com todas as tarefas tediosas, como gerar capturas de tela, lidar com a assinatura de código e lançar seu aplicativo.

Pré-requisitos

Para concluir este tutorial, você deve primeiro configurar o seguinte:

- Um Conta da AWS
- Uma [conta de desenvolvedor da Apple](#)
- Um bucket S3 para armazenar certificados
- fastlane instalado em seu projeto - [Guia](#) para instalar o fastlane

Etapa 1: configurar o Fastlane Match com o S3 em sua máquina local

O [Fastlane Match](#) é uma das [ferramentas da Fastlane](#) e permite uma configuração perfeita para assinatura de código em seu ambiente de desenvolvimento local e em outros ambientes. CodeBuild O Fastlane Match armazena todos os seus certificados de assinatura de código e perfis de provisionamento em um Git repository/S3 Bucket/Google Cloud Storage e baixa e instala os certificados e perfis necessários quando necessário.

Neste exemplo de configuração, você configurará e usará um bucket Amazon S3 para armazenamento.

1. Inicialize a partida em seu projeto:

```
fastlane match init
```

2. Quando solicitado, escolha S3 como modo de armazenamento.
3. Atualize seu `Matchfile` para usar o S3:

```
storage_mode("s3")
s3_bucket("your-s3-bucket-name")
s3_region("your-aws-region")
```

```
type("appstore") # The default type, can be: appstore, adhoc, enterprise or development
```

Etapa 2: configurar seu Fastfile

Crie ou atualize seu `Fastfile` com a seguinte faixa.

Ativado CodeBuild, o Fastlane Match precisará ser executado toda vez que você criar e assinar seu aplicativo. A maneira mais fácil de fazer isso é adicionar a match ação à pista que cria seu aplicativo.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

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

Note

Certifique-se de adicionar `setup_ci` à `before_all` seção Fastfile para que a ação de partida funcione corretamente. Isso garante que um chaveiro Fastlane temporário com as permissões apropriadas seja usado. Sem usar isso, você pode ver falhas na construção ou resultados inconsistentes.

Etapa 3: Execute o **fastlane match** comando para gerar os respectivos certificados e perfis

O comando `fastlane match` para o tipo especificado (ou seja, `development`, `appstore`, `adhoc`, `enterprise`) gerará o certificado e o perfil se não estiver disponível na loja remota. Os certificados e perfis serão armazenados no S3 pela `fastlane`.

```
bundle exec fastlane match appstore
```

A execução do comando será interativa e o `fastlane` solicitará a definição da senha para descriptografar os certificados.

Etapa 4: Crie o arquivo do aplicativo para seu projeto

Crie ou adicione o arquivo do aplicativo conforme apropriado para seu projeto.

1. Crie ou adicione o [Gymfile](#), [Appfile](#), [Snapfile](#), [Deliverfile](#) com base nos requisitos de construção do seu projeto.
2. Confirme as alterações em seu repositório remoto

Etapa 5: Criar variáveis de ambiente no Secrets Manager

Crie dois segredos para armazenar o cookie da sessão `fastlane` e a frase secreta correspondente. Para obter mais informações sobre a criação de segredos no Secrets Manager, consulte [Criar um AWS Secrets Manager segredo](#).

1. Acesse seu cookie de sessão `fastlane` da seguinte forma.
 - a. Chave secreta - `FASTLANE_SESSION`
 - b. Valor secreto - cookie de sessão gerado pela execução do seguinte comando em sua máquina local.

Note

Esse valor está disponível após a autenticação em um arquivo local: `~/.fastlane/spaceship/my_appleid_username/cookie`.

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

2. Frase secreta do Fastlane Match - Para permitir que o Fastlane Match decifre os certificados e perfis armazenados no bucket do S3, é necessário adicionar a senha de criptografia que você configurou na etapa de configuração do Match às variáveis de ambiente do projeto. CodeBuild
 - a. Chave secreta - MATCH_PASSWORD
 - b. Valor secreto - *<match passphrase to decrypt certificates>*. A frase secreta é definida ao gerar os certificados na Etapa 3.

Note

Ao criar os segredos acima no Secrets Manager, lembre-se de fornecer um nome secreto com o seguinte prefixo: /CodeBuild/

Etapa 6: criar uma frota de computação

Crie a frota de computação para seu projeto.

1. No console, acesse CodeBuild e crie uma nova frota de computação.
2. Escolha “macOS” como sistema operacional e selecione um tipo de computação e uma imagem apropriados.

Etapa 7: criar um projeto no CodeBuild

Crie seu projeto em CodeBuild.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
3. Configure seu provedor de origem (como GitHub, CodeCommit). Este é o repositório de origem do projeto iOS e não o repositório de certificados.

4. Em Ambiente:

- Escolha Capacidade reservada.
- Em Frota, selecione a frota criada acima.
- Forneça o nome da função de serviço que CodeBuild será criada para você.
- Forneça as variáveis de ambiente abaixo.
 - Nome: MATCH_PASSWORD, Valor: `<secrets arn>`, Tipo: Secrets Manager (Secrets ARN criado na etapa 5 para MATCH_PASSWORD)
 - Nome: FASTLANE_SESSION, Valor: `<secrets arn>`, Tipo: Secrets Manager (Secrets ARN criado na Etapa 5 para FASTLANE_SESSION)

5. No Buildspec, adicione o seguinte:

```
version: 0.2

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

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

Etapa 8: configurar a função do IAM

Depois que o projeto for criado, certifique-se de que a função de serviço do seu CodeBuild projeto tenha permissões para acessar o bucket do S3 que contém os certificados. Adicione a seguinte política à função:

```
{
```

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

Etapa 9: executar a compilação

Execute o build. Você pode revisar o status da compilação e fazer login CodeBuild.

Depois que o trabalho for concluído, você poderá visualizar o log do trabalho.

Solução de problemas

- Se você encontrar problemas com a busca de certificados, certifique-se de que suas permissões do IAM estejam configuradas corretamente para acesso ao S3.
- Se você encontrar problemas com a decodificação do certificado, certifique-se de definir a senha correta na variável de ambiente MATCH_PASSWORD.
- Para problemas de assinatura de código, verifique se sua conta de desenvolvedor da Apple tem os certificados e perfis necessários e se o identificador do pacote no seu projeto do Xcode corresponde ao do seu perfil de provisionamento.

Considerações sobre segurança

A seguir estão as considerações de segurança para este tutorial.

- Certifique-se de que seu bucket do S3 tenha as configurações de segurança adequadas, incluindo criptografia em repouso. Em particular, certifique-se de que o bucket não tenha acesso público e restrinja o acesso somente CodeBuild ao sistema que precisa ter acesso.
- Considere usar AWS Secrets Manager para armazenar informações confidenciais, como `MATCH_PASSWORD` e `FASTLANE_SESSION`.

Este exemplo fornece uma configuração para assinatura de código iOS com Fastlane CodeBuild usando o Amazon S3 para armazenamento de certificados. Talvez seja necessário ajustar algumas etapas com base nos requisitos e no CodeBuild ambiente específicos do projeto. Essa abordagem aproveita os AWS serviços para aumentar a segurança e a integração no AWS ecossistema.

Tutorial: Assinatura de código da Apple com Fastlane em CodeBuild uso GitHub para armazenamento de certificados

[fastlane](#) é uma ferramenta popular de automação de código aberto para automatizar implantações e lançamentos beta para seus aplicativos iOS e Android. Ele lida com todas as tarefas tediosas, como gerar capturas de tela, lidar com a assinatura de código e lançar seu aplicativo.

Este exemplo demonstra como configurar a assinatura de código da Apple usando o Fastlane em um CodeBuild projeto executado no Mac Fleet, com GitHub armazenamento para certificados e perfis de provisionamento.

Pré-requisitos

Para concluir este tutorial, você deve primeiro configurar o seguinte:

- Um Conta da AWS
- Uma [conta de desenvolvedor da Apple](#)
- Um GitHub repositório privado para armazenar certificados
- fastlane instalado em seu projeto - [Guia](#) para instalar o fastlane

Etapa 1: Configure o Fastlane Match GitHub em sua máquina local

O [Fastlane Match](#) é uma das [ferramentas da Fastlane](#) e permite uma configuração perfeita para assinatura de código em seu ambiente de desenvolvimento local e em outros ambientes. CodeBuild O Fastlane Match armazena todos os seus certificados de assinatura de código e perfis

de provisionamento em um Git repository/S3 Bucket/Google Cloud Storage e baixa e instala os certificados e perfis necessários quando necessário.

Neste exemplo de configuração, vamos configurar e usar um repositório Git para armazenamento.

1. Inicialize a partida em seu projeto:

```
fastlane match init
```

2. Quando solicitado, escolha GitHub como modo de armazenamento.
3. Atualize seu `Matchfile` para usar GitHub:

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

Note

Certifique-se de inserir o URL HTTPS do seu repositório Git para fastlane para autenticar e clonar com sucesso. Caso contrário, você poderá ver um erro de autenticação ao tentar usar o match.

Etapa 2: configurar seu Fastfile

Crie ou atualize seu `Fastfile` com a seguinte faixa.

Ativado CodeBuild, o Fastlane Match precisará ser executado toda vez que você criar e assinar seu aplicativo. A maneira mais fácil de fazer isso é adicionar a `match` ação à pista que cria seu aplicativo.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end
end
```



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

Note

Certifique-se de adicionar `setup_ci` à `before_all` seção Fastfile para que a ação de partida funcione corretamente. Isso garante que um chaveiro Fastlane temporário com as permissões apropriadas seja usado. Sem usar isso, você pode ver falhas na construção ou resultados inconsistentes.

Etapa 3: Execute o **fastlane match** comando para gerar os respectivos certificados e perfis

O comando `fastlane match` para o tipo especificado (ou seja, `development`, `appstore`, `adhoc`, `enterprise`) gerará o certificado e o perfil se não estiver disponível na loja remota. Os certificados e perfis serão armazenados no GitHub pela `fastlane`.

```
bundle exec fastlane match appstore
```

A execução do comando será interativa e o `fastlane` solicitará a definição de uma senha para descriptografar os certificados.

Etapa 4: Crie o arquivo do aplicativo para seu projeto

Crie ou adicione o arquivo do aplicativo conforme apropriado para seu projeto.

1. Crie ou adicione o [Gymfile](#), [Appfile](#), [Snapfile](#), [Deliverfile](#) com base nos requisitos de construção do seu projeto.
2. Confirme as alterações no seu repositório remoto.

Etapa 5: Criar variáveis de ambiente no Secrets Manager

Crie três segredos para armazenar o cookie da sessão fastlane e a senha correspondente. Para obter mais informações sobre a criação de segredos no Secrets Manager, consulte [Criar um AWS Secrets Manager segredo](#).

1. Acesse seu cookie de sessão fastlane da seguinte forma.
 - a. Chave secreta - FASTLANE_SESSION
 - b. Valor secreto - cookie de sessão gerado pela execução do seguinte comando em sua máquina local.

 Note

Esse valor está disponível após a autenticação em um arquivo local: `~/fastlane/spaceship/my_appleid_username/cookie`.

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

2. Frase secreta do Fastlane Match - Para permitir que o Fastlane Match decifre os certificados e perfis armazenados no repositório Git, é necessário adicionar a senha de criptografia que você configurou na etapa de configuração do Match às variáveis de ambiente do projeto. CodeBuild
 - a. Chave secreta - MATCH_PASSWORD
 - b. Valor secreto - `<match passphrase to decrypt certificates>`. A frase secreta é definida ao gerar os certificados na Etapa 3.
3. Fastlane MATCH_GIT_BASIC_AUTHORIZATION - defina uma autorização básica para a partida:
 - a. Chave secreta:

MATCH_GIT_BASIC_AUTHORIZATION
 - b. Valor secreto - O valor deve ser uma string codificada em base64 do seu nome de usuário e token de acesso pessoal (PAT) no formato. `username:password` Você pode gerá-lo usando o seguinte comando:

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

Você pode gerar seu PAT no GitHub console em Seu perfil > Configurações > Configurações de desenvolvedores > Token de acesso pessoal. Para obter mais informações, consulte o guia a seguir: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>.

 Note

Ao criar os segredos acima no Secrets Manager, lembre-se de fornecer um nome secreto com o seguinte prefixo: `/CodeBuild/`

Etapa 6: criar uma frota de computação

Crie a frota de computação para seu projeto.

1. No console, acesse CodeBuild e crie uma nova frota de computação.
2. Escolha macOS como sistema operacional e selecione um tipo de computação e uma imagem apropriados.

Etapa 7: criar um projeto no CodeBuild

Crie seu projeto em CodeBuild.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
3. Configure seu provedor de origem (como GitHub, CodeCommit). Este é o repositório de origem do projeto iOS e não o repositório de certificados.
4. Em Ambiente:
 - Escolha Capacidade reservada.
 - Em Frota, selecione a frota criada acima.
 - Forneça o nome da função de serviço que CodeBuild será criada para você.
 - Forneça as variáveis de ambiente abaixo.

- Nome: MATCH_PASSWORD, Valor: `<secrets arn>`, Tipo: Secrets Manager (Secrets ARN criado na etapa 5 para MATCH_PASSWORD)
- Nome: FASTLANE_SESSION, Valor: `<secrets arn>`, Tipo: Secrets Manager (Secrets ARN criado na etapa 5 para FASTLANE_SESSION)
- Nome: MATCH_GIT_BASIC_AUTHORIZATION, Valor: `<secrets ARN>`, Tipo: Secrets Manager Secrets ARN (criado na etapa 5 para) MATCH_GIT_BASIC_AUTHORIZATION

5. No Buildspec, adicione o seguinte:

```
version: 0.2

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

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

Etapa 8: executar a compilação

Execute o build. Você pode revisar o status da compilação e fazer login CodeBuild.

Depois que o trabalho for concluído, você poderá visualizar o log do trabalho.

Solução de problemas

- Se você tiver problemas para acessar o GitHub repositório, verifique novamente seu token de acesso pessoal e a variável de ambiente MATCH_GIT_BASIC_AUTHORIZATION.

- Se você encontrar problemas com a decodificação do certificado, certifique-se de definir a senha correta na variável de ambiente `MATCH_PASSWORD`.
- Para problemas de assinatura de código, verifique se sua conta de desenvolvedor da Apple tem os certificados e perfis necessários e se o identificador do pacote no seu projeto do Xcode corresponde ao do seu perfil de provisionamento.

Considerações sobre segurança

A seguir estão as considerações de segurança para este tutorial.

- Mantenha seu GitHub repositório de certificados privado e audite regularmente o acesso.
- Considere usar AWS Secrets Manager para armazenar informações confidenciais, como `MATCH_PASSWORD` e `FASTLANE_SESSION`.

Este exemplo fornece uma configuração para a assinatura de código do iOS com o Fastlane e CodeBuild o uso GitHub para armazenamento de certificados. Talvez seja necessário ajustar algumas etapas com base nos requisitos e no CodeBuild ambiente específicos do projeto. Essa abordagem aproveita os AWS serviços para aumentar a segurança e a integração no AWS ecossistema.

Defina os nomes dos artefatos no momento da compilação usando versionamento semântico

Este exemplo contém exemplos de arquivos buildspec que demonstram como especificar o nome de um artefato que é criado no momento da compilação. Um nome especificado em um arquivo buildspec pode incorporar comandos Shell e variáveis de ambiente para torná-lo exclusivo. Um nome especificado em um arquivo buildspec substitui um nome inserido no console quando você cria o projeto.

Se você compilar várias vezes, o uso do nome de um artefato especificado no arquivo buildspec poderá garantir que os nomes dos arquivos de artefato de saída sejam exclusivos. Por exemplo, você pode usar uma data e um time stamp que é inserido no nome de um artefato no momento da compilação.

Se você quiser substituir o nome do artefato que inseriu no console por um nome no arquivo buildspec, faça o seguinte:

1. Defina o projeto de compilação para substituir o nome do artefato por um nome no arquivo buildspec.
 - Se você usar o console do para criar seu projeto de compilação, selecione Enable semantic versioning (Habilitar versionamento semântico). Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#).
 - Se você usar o AWS CLI, defina o como overrideArtifactName true no arquivo formatado em JSON passado para. create-project Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).
 - Se você usa a AWS CodeBuild API, defina a overrideArtifactName bandeira no ProjectArtifacts objeto quando um projeto for criado ou atualizado ou quando uma compilação for iniciada.
2. Especifique um nome no arquivo buildspec. Use os seguintes exemplos de arquivos buildspec como um guia.

Este exemplo do Linux mostra como especificar um artefato que inclui a data em que a compilação foi criada:

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${date +%Y-%m-%d}
```

Este exemplo do Linux mostra como especificar o nome de um artefato que usa uma variável de ambiente do CodeBuild. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
```

```
- '**/*'
name: myname-$AWS_REGION
```

Este exemplo do Windows mostra como especificar um artefato que inclui a data e a hora em que a compilação foi criada:

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Este exemplo do Windows mostra como especificar um nome de artefato que usa uma variável declarada no arquivo buildspec e uma variável de ambiente. CodeBuild Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Para obter mais informações, consulte [Referência de especificação de construção para CodeBuild](#).

Execute amostras do Microsoft Windows para CodeBuild

Esses exemplos usam um ambiente de AWS CodeBuild compilação executando o Microsoft Windows Server 2019, o .NET Framework e o SDK do .NET Core para criar arquivos de tempo de execução a partir do código escrito em F# e Visual Basic.

Important

A execução dessas amostras pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças por CodeBuild e por AWS recursos e ações relacionados ao Amazon S3 e CloudWatch aos AWS KMS registros. Para obter mais informações, consulte [CodeBuild preços, preços](#) do [Amazon S3, preços](#) e [AWS Key Management Service preços](#) da [Amazon CloudWatch](#).

Executar os exemplos do Windows

Use o procedimento a seguir para executar os exemplos do Windows.

Para executar os exemplos do Windows

1. Crie os arquivos conforme descrito nas [Arquivos](#) seções [Estrutura de diretório](#) e deste tópico e, em seguida, carregue-os em um bucket de entrada do S3 ou em um GitHub repositório CodeCommit ou.

Important

Não faça upload de *(root directory name)*, apenas dos arquivos dentro de *(root directory name)*.

Se você estiver usando um bucket de entrada do S3, crie um arquivo ZIP que contenha os arquivos e envie-o para o bucket de entrada. Não adicione *(root directory name)* ao arquivo ZIP, apenas aos arquivos dentro de *(root directory name)*.

2. Crie um projeto de compilação. O projeto de compilação deve usar a imagem `mcr.microsoft.com/dotnet/framework/sdk:4.8` para criar projetos do .NET Framework.

Se você usar o AWS CLI para criar o projeto de construção, a entrada formatada em JSON para o `create-project` comando poderá ser semelhante a essa. (Substitua os espaços reservados com seus próprios valores.)

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Execute a compilação e siga as etapas em [Executar compilações manualmente](#).
4. Para obter o artefato de saída da compilação, no bucket de saída do S3, faça download do arquivo *windows-build-output-artifact*.zip no computador ou na instância local. Extraia o conteúdo para obter o runtime e outros arquivos.
 - O arquivo de runtime do exemplo escrito em F# que usa o .NET Framework, `FSharpHelloWorld.exe`, pode ser encontrado no diretório `FSharpHelloWorld\bin\Debug`.
 - O arquivo de runtime do exemplo escrito em Visual Basic que usa o .NET Framework, `VBHelloWorld.exe`, pode ser encontrado no diretório `VBHelloWorld\bin\Debug`.

Estrutura de diretório

Essas amostras apresentam as seguintes estruturas de diretório.

F# e o .NET Framework

```
(root directory name)
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
    ### App.config
    ### AssemblyInfo.fs
    ### FSharpHelloWorld.fsproj
    ### Program.fs
```

Visual Basic e o .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
    ### App.config
    ### HelloWorld.vb
    ### VBHelloWorld.vbproj
    ### My Project
        ### Application.Designer.vb
        ### Application.myapp
        ### AssemblyInfo.vb
        ### Resources.Designer.vb
        ### Resources.resx
        ### Settings.Designer.vb
        ### Settings.settings
```

Arquivos

Esses exemplos usam os arquivos a seguir.

F# e o .NET Framework

buildspec.yml (em *(root directory name)*):

```

version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln (em *(root directory name)*):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config (em *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

AssemblyInfo.fs (em *(root directory name)*\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
```

```
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]

do
    ()
```

FSharpHelloWorld.fsproj (em *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
    Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <Tailcalls>>false</Tailcalls>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
```

```

</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <DebugType>pdbonly</DebugType>
  <Optimize>>true</Optimize>
  <Tailcalls>>true</Tailcalls>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <WarningLevel>3</WarningLevel>
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
  <Prefer32Bit>>true</Prefer32Bit>
</PropertyGroup>
<ItemGroup>
  <Reference Include="mscorlib" />
  <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
    <Private>True</Private>
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Core" />
  <Reference Include="System.Numerics" />
</ItemGroup>
<ItemGroup>
  <Compile Include="AssemblyInfo.fs" />
  <Compile Include="Program.fs" />
  <None Include="App.config" />
</ItemGroup>
<PropertyGroup>
  <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''">11</
MinimumVisualStudioVersion>
</PropertyGroup>
<Choose>
  <When Condition="'$(VisualStudioVersion)' == '11.0'">
    <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
    </PropertyGroup>
  </When>
  <Otherwise>
    <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>

```

```

    </PropertyGroup>
  </Otherwise>
</Choose>
<Import Project="$(FSharpTargetsPath)" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>

```

Program.fs (em *(root directory name)*\FSharpHelloWorld):

```

// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code

```

Visual Basic e o .NET Framework

buildspec.yml (em *(root directory name)*):

```

version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
        PackagesDirectory $env:PACKAGE_DIRECTORY'

```

```
- '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\VBHelloWorld\bin\Debug\*
```

VBHelloWorld.sln (em *(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld
\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config (em *(root directory name)\VBHelloWorld*):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

HelloWorld.vb (em *(root directory name)\VBHelloWorld*):

Module HelloWorld

```
Sub Main()
    MsgBox("Hello World")
End Sub
```

```
End Module
```

VBHelloWorld.vbproj (em *(root directory name)\VBHelloWorld*):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
    Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
```

```
<DefineDebug>false</DefineDebug>
<DefineTrace>true</DefineTrace>
<Optimize>true</Optimize>
<OutputPath>bin\Release\</OutputPath>
<DocumentationFile>VBHelloWorld.xml</DocumentationFile>
<NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
</PropertyGroup>
<PropertyGroup>
  <OptionExplicit>On</OptionExplicit>
</PropertyGroup>
<PropertyGroup>
  <OptionCompare>Binary</OptionCompare>
</PropertyGroup>
<PropertyGroup>
  <OptionStrict>Off</OptionStrict>
</PropertyGroup>
<PropertyGroup>
  <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Xml" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
  <Import Include="Microsoft.VisualBasic" />
  <Import Include="System" />
  <Import Include="System.Collections" />
  <Import Include="System.Collections.Generic" />
  <Import Include="System.Data" />
  <Import Include="System.Diagnostics" />
  <Import Include="System.Linq" />
  <Import Include="System.Xml.Linq" />
  <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
  <Compile Include="HelloWorld.vb" />
  <Compile Include="My Project\AssemblyInfo.vb" />
  <Compile Include="My Project\Application.Designer.vb">
```

```

    <AutoGen>True</AutoGen>
    <DependentUpon>Application.myapp</DependentUpon>
</Compile>
<Compile Include="My Project\Resources.Designer.vb">
    <AutoGen>True</AutoGen>
    <DesignTime>True</DesignTime>
    <DependentUpon>Resources.resx</DependentUpon>
</Compile>
<Compile Include="My Project\Settings.Designer.vb">
    <AutoGen>True</AutoGen>
    <DependentUpon>Settings.settings</DependentUpon>
    <DesignTimeSharedInput>True</DesignTimeSharedInput>
</Compile>
</ItemGroup>
<ItemGroup>
    <EmbeddedResource Include="My Project\Resources.resx">
        <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
        <LastGenOutput>Resources.Designer.vb</LastGenOutput>
        <CustomToolNamespace>My.Resources</CustomToolNamespace>
        <SubType>Designer</SubType>
    </EmbeddedResource>
</ItemGroup>
<ItemGroup>
    <None Include="My Project\Application.myapp">
        <Generator>MyApplicationCodeGenerator</Generator>
        <LastGenOutput>Application.Designer.vb</LastGenOutput>
    </None>
    <None Include="My Project\Settings.settings">
        <Generator>SettingsSingleFileGenerator</Generator>
        <CustomToolNamespace>My</CustomToolNamespace>
        <LastGenOutput>Settings.Designer.vb</LastGenOutput>
    </None>
    <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.

    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->

```

```
</Project>
```

Application.Designer.vb (em *(root directory name)*\VBHelloWorld\My Project):

```
' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

Option Strict On
Option Explicit On
```

Application.myapp (em *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb (em *(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes
```

```

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>

'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>

```

Resources.Designer.vb (em *(root directory name)*\VBHelloWorld\My Project):

```

'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On

Namespace My.Resources

```

```

'This class was auto-generated by the StronglyTypedResourceBuilder
'class via a tool like ResGen or Visual Studio.
'To add or remove a member, edit your .ResX file then rerun ResGen
'with the /str option, or rebuild your VS project.
'''<summary>
'''  A strongly-typed resource class, for looking up localized strings, etc.
'''</summary>

<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.Microsoft.VisualBasic.HideModuleNameAttribute(> _
Friend Module Resources

    Private resourceMan As Global.System.Resources.ResourceManager

    Private resourceCulture As Global.System.Globalization.CultureInfo

    '''<summary>
    '''  Returns the cached ResourceManager instance used by this class.
    '''</summary>

    <Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
    Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
        Get
            If Object.ReferenceEquals(resourceMan, Nothing) Then
                Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
                resourceMan = temp
            End If
            Return resourceMan
        End Get
    End Property

    '''<summary>
    '''  Overrides the current thread's CurrentUICulture property for all
    '''  resource lookups using this strongly typed resource class.
    '''</summary>

```

```

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
    Friend Property Culture() As Global.System.Globalization.CultureInfo
        Get
            Return resourceCulture
        End Get
        Set(ByVal value As Global.System.Globalization.CultureInfo)
            resourceCulture = value
        End Set
    End Property
End Module
End Namespace

```

Resources.resx (em *(root directory name)\VBHelloWorld\My Project*):

```

<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema

    Version 2.0

    The primary goals of this format is to allow a simple XML format
    that is mostly human readable. The generation and parsing of the
    various data types are done through the TypeConverter classes
    associated with the data types.

    Example:

    ... ado.net/XML headers & schema ...
    <resheader name="resmimetype">text/microsoft-resx</resheader>
    <resheader name="version">2.0</resheader>
    <resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
    <resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
    <data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
    <data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
    <data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
      <value>[base64 mime encoded serialized .NET Framework object]</value>
    </data>

```

```

<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
  <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
  <comment>This is a comment</comment>
</data>

```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

```

mimetype: application/x-microsoft.net.object.binary.base64
value    : The object must be serialized with
           : System.Serialization.Formatters.Binary.BinaryFormatter
           : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.soap.base64
value    : The object must be serialized with
           : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
           : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.bytearray.base64
value    : The object must be serialized into a byte array
           : using a System.ComponentModel.TypeConverter
           : and then encoded with base64 encoding.

```

```
-->
```

```

<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">

```



```

<xsd:element name="metadata">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" />
    <xsd:attribute name="type" type="xsd:string" />
    <xsd:attribute name="mimetype" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="assembly">
  <xsd:complexType>
    <xsd:attribute name="alias" type="xsd:string" />
    <xsd:attribute name="name" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="data">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="resheader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
  <value>text/microsoft-resx</value>

```

```

</resheader>
<resheader name="version">
  <value>2.0</value>
</resheader>
<resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
</root>

```

Settings.Designer.vb (em *(root directory name)*\VBHelloWorld\My Project):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

```

```

Option Strict On
Option Explicit On

```

```

Namespace My

```

```

    <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _

```

```

    Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
    "11.0.0.0"), _

```

```

    Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrows

```

```

    -

```

```

    Partial Friend NotInheritable Class MySettings

```

```

        Inherits Global.System.Configuration.ApplicationSettingsBase

```

```

    Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
MySettings), MySettings)

#Region "My.Settings Auto-Save Functionality"
    #If _MyType = "WindowsForms" Then
        Private Shared addedHandler As Boolean

        Private Shared addedHandlerLockObject As New Object

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
        Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
            If My.Application.SaveMySettingsOnExit Then
                My.Settings.Save()
            End If
        End Sub
    #End If
#End Region

Public Shared ReadOnly Property [Default]() As MySettings
    Get

        #If _MyType = "WindowsForms" Then
            If Not addedHandler Then
                SyncLock addedHandlerLockObject
                    If Not addedHandler Then
                        AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                        addedHandler = True
                    End If
                End SyncLock
            End If
        #End If
        Return defaultInstance
    End Get
End Property
End Class
End Namespace

Namespace My

    <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _

```

```
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _  
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute())> _  
Friend Module MySettingsProperty  
  
    <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _  
    Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings  
        Get  
            Return Global.VBHelloWorld.My.MySettings.Default  
        End Get  
    End Property  
End Module  
End Namespace
```

Settings.settings (em *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version='1.0' encoding='utf-8'?>  
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"  
    CurrentProfile="(Default)" UseMySettingsClassName="true">  
    <Profiles>  
        <Profile Name="(Default)" />  
    </Profiles>  
    <Settings />  
</SettingsFile>
```

Planeje uma incorporação AWS CodeBuild

Antes de usar AWS CodeBuild, você deve responder a estas perguntas:

1. Onde o código-fonte está armazenado? CodeBuild atualmente suporta a construção a partir dos seguintes provedores de repositórios de código-fonte. O código-fonte deve conter um arquivo de especificação de compilação (buildspec). Um buildspec é uma coleção de comandos de compilação e configurações relacionadas, no formato YAML, CodeBuild usados para executar uma compilação. É possível declarar um buildspec em uma definição de projeto de compilação.

Provedor do repositório	Obrigatório	Documentação
CodeCommit	Nome do repositório. (Opcional) Commit ID associada ao código-fonte.	Consulte estes tópicos no Guia do usuário do AWS CodeCommit : Crie um CodeCommit repositório Crie um commit em CodeCommit
Amazon S3	Nome do bucket de entrada. Nome do objeto correspondente ao arquivo ZIP de entrada do build que contém o código-fonte. (Opcional) Version ID associada ao	Consulte estes tópicos no Guia de noções básicas do Amazon S3: Criar um bucket Adicionar um objeto a um bucket

Provedor do repositório	Obrigatório	Documentação
	arquivo ZIP de entrada do build.	
GitHub	Nome do repositório. (Opcional) Commit ID associada ao código-fonte.	Veja este tópico no site de GitHub Ajuda: Create a repo
Bitbucket	Nome do repositório. (Opcional) Commit ID associada ao código-fonte.	Consulte este tópico no site de documentação do Bitbucket Cloud: Criar um repositório

2. Que comandos de build você precisa executar e em que ordem? Por padrão, CodeBuild baixa a entrada de compilação do provedor especificado e carrega a saída da compilação para o bucket especificado. Você usa a buildspec para instruir como a entrada de compilação obtida por download é convertida na saída de compilação esperada. Para obter mais informações, consulte o [Referência de buildspec](#).
3. De que runtimes e ferramentas você precisa para executar o build? Por exemplo, você está fazendo um build para Java, Ruby, Python ou Node.js? O build precisa de Maven ou Ant, ou de um compilador para Java, Ruby ou Python? A compilação precisa do Git AWS CLI, do ou de outras ferramentas?

CodeBuild executa compilações em ambientes de compilação que usam imagens do Docker. Essas imagens Docker devem ser armazenadas em um tipo de repositório suportado por CodeBuild. Isso inclui o repositório de imagens do CodeBuild Docker, o Docker Hub e o Amazon Elastic Container Registry (Amazon ECR). Para obter mais informações sobre o repositório de imagens do CodeBuild Docker, consulte [Imagens do Docker fornecidas por CodeBuild](#)

4. Você precisa de AWS recursos que não sejam fornecidos automaticamente pelo CodeBuild? Em caso afirmativo, de quais políticas de segurança esses recursos precisam? Por exemplo, talvez seja necessário modificar a função CodeBuild de serviço para CodeBuild permitir trabalhar com esses recursos.
5. Você quer CodeBuild trabalhar com sua VPC? Nesse caso, você precisará do ID da VPC, da sub-rede IDs e do grupo de segurança para IDs sua configuração de VPC. Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

Depois de responder a essas perguntas, você deverá ter as configurações e recursos necessários para executar com êxito um build. Para executar seu build, você pode:

- Use o AWS CodeBuild console, AWS CLI, ou AWS SDKs. Para obter mais informações, consulte [Executar compilações manualmente](#).
- Crie ou identifique um pipeline e AWS CodePipeline, em seguida, adicione uma ação de compilação ou teste que CodeBuild instrua a testar automaticamente seu código, executar sua compilação ou ambas. Para obter mais informações, consulte [Use CodeBuild com CodePipeline](#).

Referência de especificação de construção para CodeBuild

Este tópico fornece informações de referência importantes sobre os arquivos de especificação de compilação (buildspec). Um buildspec é uma coleção de comandos de compilação e configurações relacionadas, no formato YAML, CodeBuild usados para executar uma compilação. É possível incluir um buildspec como parte do código-fonte ou defini-lo ao criar um projeto de compilação. Para obter informações sobre como uma build spec funciona, consulte [Como CodeBuild funciona](#).

Tópicos

- [Nome do arquivo buildspec e local de armazenamento](#)
- [Sintaxe de buildspec](#)
- [Exemplo de buildspec](#)
- [Versões de buildspec](#)
- [Referência de buildspec de compilação em lote](#)

Nome do arquivo buildspec e local de armazenamento

Se você incluir uma especificação de compilação como parte do código-fonte, por padrão, o arquivo buildspec deverá se chamar `buildspec.yml` e ser colocado na raiz do diretório de origem.

É possível substituir o nome e o local do arquivo buildspec padrão. Por exemplo, é possível:

- Use um arquivo buildspec diferente para compilações diferentes no mesmo repositório, como `buildspec_debug.yml` e `buildspec_release.yml`.
- Armazene um arquivo buildspec em um local que não seja a raiz de seu diretório de origem, como `config/buildspec.yml`, ou em um bucket do S3. O bucket do S3 deve estar na mesma AWS região do seu projeto de compilação. Especifique o arquivo buildspec usando seu ARN (por exemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

É possível especificar somente uma buildspec para um projeto de compilação, independentemente do nome do arquivo buildspec.

Para substituir o nome do arquivo buildspec padrão, o local ou ambos, faça o seguinte:

- Execute o `update-project` comando AWS CLI `create-project` or, definindo o buildspec valor do caminho para o arquivo buildspec alternativo em relação ao valor da variável de ambiente

integrada. `CODEBUILD_SRC_DIR` Você também pode fazer o equivalente com a `create project` operação no AWS SDKs. Para ter mais informações, consulte [Criar um projeto de compilação](#) ou [Alterar as configurações do projeto de compilação](#).

- Execute o AWS CLI `start-build` comando, definindo o `buildspecOverride` valor do caminho para o arquivo `buildspec` alternativo em relação ao valor da variável de ambiente integrada. `CODEBUILD_SRC_DIR` Você também pode fazer o equivalente com a `start build` operação no AWS SDKs. Para obter mais informações, consulte [Executar compilações manualmente](#).
- Em um AWS CloudFormation modelo, defina a `BuildSpec` propriedade de `Source` em um recurso do tipo `AWS::CodeBuild::Project` para o caminho para o arquivo `buildspec` alternativo em relação ao valor da variável de ambiente integrada. `CODEBUILD_SRC_DIR` Para obter mais informações, consulte a `BuildSpec` propriedade na [fonte AWS CodeBuild do projeto](#) no Guia AWS CloudFormation do usuário.

Sintaxe de buildspec

Os arquivos `buildspec` devem ser expressos no formato [YAML](#).

Se um comando contiver um caractere ou uma string de caracteres que não seja compatível com o YAML, coloque o comando entre aspas (`""`). O comando a seguir está entre aspas porque um dois pontos (`:`) seguido por um espaço não é permitido no YAML. As aspas no comando são escapadas (`\`).

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\",,]//g')"
```

A `buildspec` possui a seguinte sintaxe:

```
version: 0.2

run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
  parameter-store:
    key: "value"
    key: "value"
```

exported-variables:

- *variable*
- *variable*

secrets-manager:

key: secret-id:json-key:version-stage:version-id

git-credential-helper: no | yes

proxy:

upload-artifacts: no | yes

logs: no | yes

batch:

fast-fail: false | true

build-list:

build-matrix:

build-graph:

build-fanout:

phases:install:

run-as: *Linux-user-name*

on-failure: ABORT | CONTINUE | RETRY | RETRY-*count* | RETRY-*regex* |

RETRY-*count-regex*

runtime-versions:

runtime: version

runtime: version

commands:

- *command*
- *command*

finally:

- *command*
- *command*

pre_build:

run-as: *Linux-user-name*

on-failure: ABORT | CONTINUE | RETRY | RETRY-*count* | RETRY-*regex* |

RETRY-*count-regex*

commands:

- *command*
- *command*

finally:

- *command*
- *command*

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

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

reports:
  report-group-name-or-arn:
  files:
    - location
    - location
  base-directory: location
  discard-paths: no | yes
  file-format: report-format
artifacts:
  files:
    - location
    - location
  name: artifact-name
  discard-paths: no | yes
  base-directory: location
  exclude-paths: excluded paths
  enable-symlinks: no | yes
  s3-prefix: prefix
  secondary-artifacts:
    artifactIdentifier:
      files:
        - location
```

```
- location
  name: secondary-artifact-name
  discard-paths: no | yes
  base-directory: location
  artifactIdentifier:
    files:
      - location
      - location
    discard-paths: no | yes
    base-directory: location
cache:
  key: key
  fallback-keys:
    - fallback-key
    - fallback-key
  action: restore | save
  paths:
    - path
    - path
```

A buildspec contém o seguinte:

version

Mapeamento necessário. Representa a versão de buildspec. Recomendamos usar o 0.2.

Note

Embora a versão 0.1 ainda tenha suporte, recomendamos que você use a versão 0.2 sempre que possível. Para obter mais informações, consulte [Versões de buildspec](#).

run-as

Sequência opcional. Disponível somente para usuários do Linux. Especifica um usuário do Linux que executa comandos nesse arquivo buildspec. O run-as concede ao usuário especificado permissões de leitura e execução. Quando você especificar run-as na parte superior do arquivo buildspec globalmente, ele se aplicará a todos os comandos. Se não quiser especificar um arquivo buildspec para todos os comandos, você pode especificar um para comandos em uma fase usando run-as em um dos blocos phases. Se run-as não for especificado, todos os comandos serão executados como raiz.

env

Sequência opcional. Representa informações para uma ou mais variáveis de ambiente personalizadas.

Note

Para proteger informações confidenciais, o seguinte está oculto nos CodeBuild registros:

- AWS chave de acesso IDs. Para obter mais informações, consulte [Managing Access Keys for IAM Users](#) no Guia do usuário do AWS Identity and Access Management .
- Strings especificadas usando o repositório de parâmetros. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.
- Cadeias de caracteres especificadas usando AWS Secrets Manager. Para obter mais informações, consulte [Gerenciamento de chaves](#).

env/shell

Sequência opcional. Especifica o shell compatível com os sistemas operacionais Linux ou Windows.

Para sistemas operacionais Linux, as tags de shell compatíveis são:

- `bash`
- `/bin/sh`

Para sistemas operacionais Windows, as tags de shell compatíveis são:

- `powershell.exe`
- `cmd.exe`

env/variables

Necessário caso `env` seja especificado e você queira definir variáveis de ambiente personalizadas em texto sem formatação. Contém um mapeamento de *key/value* escalars, em que cada mapeamento representa uma única variável de ambiente personalizada em texto simples. *key* é o nome da variável de ambiente personalizada e *value* é o valor dessa variável.

Important

Não recomendamos o armazenamento de valores confidenciais em variáveis de ambiente. As variáveis de ambiente podem ser exibidas em texto simples usando ferramentas como o CodeBuild console e AWS CLI. Para valores confidenciais, recomendamos usar o mapeamento `parameter-store` ou `secrets-manager`, conforme descrito posteriormente nesta seção.

Qualquer variável de ambiente definida por você substitui variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você definir uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já contiver uma variável de ambiente chamada `PATH` com um valor de `/usr/local/sbin:/usr/local/bin` e você definir uma variável de ambiente chamada `PATH` com um valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`.

Não defina nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome é definida em vários locais, o valor será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade. Você pode adicionar ou substituir variáveis de ambiente ao criar uma compilação. Para obter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#).
- O valor na definição de projeto de build tem a precedência seguinte. Você pode adicionar variáveis de ambiente no nível do projeto ao criar ou editar um projeto. Para obter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) e [Alterar as configurações do projeto de construção em AWS CodeBuild](#).
- O valor na declaração de `buildspec` tem a menor prioridade.

env/parameter-store

Obrigatório se `env` for especificado e você quiser recuperar variáveis de ambiente personalizadas armazenadas no Amazon EC2 Systems Manager Parameter Store. Contém um mapeamento de `key/value` scalars, em que cada mapeamento representa uma única variável de ambiente personalizada armazenada no Amazon EC2 Systems Manager Parameter Store. `key` é o nome

que você usa posteriormente em seus comandos de compilação para se referir a essa variável de ambiente personalizada e *value* é o nome da variável de ambiente personalizada armazenada no Amazon EC2 Systems Manager Parameter Store. Para armazenar valores confidenciais, consulte [Armazenamento de parâmetros e instruções passo a passo do Systems Manager: Crie e teste um parâmetro de string \(console\) no Guia](#) do usuário do Amazon EC2 Systems Manager.

Important

CodeBuild Para permitir a recuperação de variáveis de ambiente personalizadas armazenadas no Amazon EC2 Systems Manager Parameter Store, você deve adicionar a `ssm:GetParameters` ação à sua função CodeBuild de serviço. Para obter mais informações, consulte [CodeBuild Permitir interagir com outros AWS serviços](#).

Todas as variáveis de ambiente recuperadas do Amazon EC2 Systems Manager Parameter Store substituem as variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você recuperar uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já contiver uma variável de ambiente chamada `PATH` com um valor de `/usr/local/sbin:/usr/local/bin` e você recuperar uma variável de ambiente chamada `PATH` com um valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`.

Não armazene nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome é definida em vários locais, o valor será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade. Você pode adicionar ou substituir variáveis de ambiente ao criar uma compilação. Para obter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#).
- O valor na definição de projeto de build tem a precedência seguinte. Você pode adicionar variáveis de ambiente no nível do projeto ao criar ou editar um projeto. Para obter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) e [Alterar as configurações do projeto de construção em AWS CodeBuild](#).
- O valor na declaração de `buildspec` tem a menor prioridade.

env/secrets-manager

Obrigatório se você quiser recuperar variáveis de ambiente personalizadas armazenadas em AWS Secrets Manager. Especifique uma `reference-key` do Secrets Manager usando o seguinte padrão:

```
<key>: <secret-id>:<json-key>:<version-stage>:<version-id>  
<key>
```

(Obrigatório) O nome da variável de ambiente local. Use esse nome para acessar a variável durante a compilação.

<secret-id>

(Obrigatório) O nome ou o nome do recurso da Amazon (ARN) que serve como identificador exclusivo para o segredo. Para acessar um segredo em sua conta da AWS , basta especificar o nome secreto. Para acessar um segredo em uma AWS conta diferente, especifique o ARN secreto.

<json-key>

(Opcional) Especifica o nome da chave do par de chave-valor cujo valor você deseja recuperar. Se você não especificar um `json-key`, CodeBuild recuperará todo o texto secreto.

<version-stage>

(Opcional) Especifica a versão do segredo que você deseja recuperar pelo rótulo temporário anexado à versão. Rótulos temporários são usados para acompanhar diferentes versões durante o processo de rodízio. Se você usar `version-stage`, não especifique `version-id`. Se você não especificar um estágio de versão ou um ID de versão, o padrão será recuperar a versão com o valor do estágio de versão de `AWSCURRENT`.

<version-id>

(Opcional) Especifica o identificador exclusivo da versão do segredo que você deseja usar. Se você especificar `version-id`, não especifique `version-stage`. Se você não especificar um estágio de versão ou um ID de versão, o padrão será recuperar a versão com o valor do estágio de versão de `AWSCURRENT`.

No exemplo a seguir, `TestSecret` é o nome do par chave-valor armazenado no Secrets Manager. A chave para `TestSecret` é `MY_SECRET_VAR`. Você acessa a variável durante a compilação usando o nome `LOCAL_SECRET_VAR`.


```
env:  
  secrets-manager:  
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Para obter mais informações, consulte [O que é o AWS Secrets Manager](#) no Guia do usuário do AWS Secrets Manager .

env/exported-variables

Mapeamento opcional. Usado para listar as variáveis de ambiente que você deseja exportar. Especifique o nome de cada variável que você deseja exportar em uma linha separada em `exported-variables`. A variável que você deseja exportar deve estar disponível no contêiner durante a compilação. A variável exportada pode ser uma variável de ambiente.

As variáveis de ambiente exportadas são usadas em conjunto com AWS CodePipeline a exportação de variáveis de ambiente do estágio de construção atual para os estágios subsequentes no pipeline. Para obter mais informações, consulte [Working with variables](#) no Guia do usuário do AWS CodePipeline .

Durante uma compilação, o valor de uma variável está disponível a partir da fase `install`. Ele pode ser atualizado entre o início da fase `install` e o final da fase `post_build`. Após o término da fase `post_build`, o valor de variáveis exportadas não pode ser alterado.

Note

Não é possível exportar o seguinte:

- Amazon EC2 Systems Manager Parameter Armazene segredos especificados no projeto de construção.
- Segredos do Secrets Manager especificados no projeto de compilação
- Variáveis de ambiente que começam com `AWS_`.

ambiente/ git-credential-helper

Mapeamento opcional. Usado para indicar se CodeBuild usa seu auxiliar de credenciais do Git para fornecer credenciais do Git. `yes` se for usado. Caso contrário, `no` ou não especificado. Para obter mais informações, consulte [gitcredentials](#) no site do Git.

Note

O `git-credential-helper` não é compatível com compilações acionadas por um webhook para um repositório Git público.

proxy

Sequência opcional. Usado para representar configurações se você executar a compilação em um servidor de proxy explícito. Para obter mais informações, consulte [Execute CodeBuild em um servidor proxy explícito](#).

proxy/upload-artifacts

Mapeamento opcional. Defina como `yes` se você quiser que a compilação em um servidor de proxy explícito faça upload de artefatos. O padrão é `no`.

proxy/logs

Mapeamento opcional. Defina como `yes` para sua construção em um servidor proxy explícito para criar CloudWatch registros. O padrão é `no`.

phases

Sequência necessária. Representa os comandos CodeBuild executados durante cada fase da compilação.

Note

Na versão 0.1 do `buildspec`, CodeBuild executa cada comando em uma instância separada do shell padrão no ambiente de compilação. Isso significa que cada comando é executado isoladamente em relação aos outros comandos. Portanto, por padrão, não é possível executar um comando que dependa do estado de algum comando anterior (por exemplo, mudança de diretórios ou configuração de variáveis de ambiente). Para resolver essa limitação, recomendamos usar a versão 0.2, que resolve o problema. Caso seja necessário usar a `buildspec` versão 0.1 por algum motivo, recomendamos as abordagens em [Shells e comandos em ambientes de compilação](#).

phases/*/run-as

Sequência opcional. Use em uma fase de compilação para especificar um usuário do Linux que executa seus comandos. Se `run-as` também for especificado globalmente para todos os comandos na parte superior do arquivo `buildspec`, o usuário em nível de fase terá precedência. Por exemplo, se `run-as` especificar globalmente `User-1` e, para a fase `install` apenas uma instrução `run-as` especificar `User-2`, todos os comandos no arquivo `buildspec` serão executados como `User-1`, exceto os comandos na fase `install`, que serão executados como `User-2`.

phases/*/on-failure

Sequência opcional. Especifica a ação a ser realizada se ocorrer uma falha durante a fase. Pode ter um dos valores a seguir:

- `ABORT`: anule a compilação.
- `CONTINUE`: vá para a próxima fase.
- `RETRY`- Tente novamente a compilação até 3 vezes com uma mensagem de erro que corresponda à expressão `. *` regular.
- `RETRY-count`- Tente novamente a compilação por um número especificado de vezes, conforme *count* representado por uma mensagem de erro que corresponda à expressão `. *` regular. Observe que *count* deve estar entre 0 e 100. Por exemplo, os valores válidos incluem `RETRY-4` `RETRY-8` e.
- `RETRY-regex`- Tente novamente a compilação até 3 vezes e use *regex* para incluir uma expressão regular que corresponda a uma mensagem de erro especificada. Por exemplo, os valores válidos incluem `Retry-.*Error: Unable to connect to database.*` `RETRY-invalid+` e.
- `RETRY-count-regex`- Repita a compilação por um determinado número de vezes, conforme representado por *count*. Observe que *count* deve estar entre 0 e 100. Você também pode usar *regex* para incluir uma expressão regular que corresponda à mensagem de erro. Por exemplo, os valores válidos incluem `Retry-3-.*connection timed out.*` `RETRY-8-invalid+` e.

Se essa propriedade não for especificada, o processo de falha seguirá as fases de transição, conforme mostrado em [Transições de fase de compilação](#).

phases/*/finally

Bloco opcional. Os comandos especificados em um bloco `finally` são executados após os comandos no bloco `commands`. Os comandos em um bloco `finally` são executados mesmo quando um comando no bloco `commands` falha. Por exemplo, se o `commands` bloco contiver

três comandos e o primeiro falhar, CodeBuild pulará os dois comandos restantes e executará qualquer comando no `finally` bloco. A fase é bem-sucedida quando todos os comandos nos blocos `commands` e `finally` são executados com êxito. Se algum comando em uma fase falhar, a fase falhará.

Os nomes permitidos para as fases de build são:

`phases/install`

Sequência opcional. Representa os comandos, se houver, que são CodeBuild executados durante a instalação. Recomendamos que você use a fase `install` somente para pacotes de instalação no ambiente de compilação. Por exemplo, você pode usar essa fase para instalar uma estrutura de teste de código, como Mocha ou RSpec.

`phases/install/runtime-versions`

Sequência opcional. Uma versão do runtime é compatível com a imagem padrão 5.0 do Ubuntu ou posterior e a imagem padrão 4.0 ou posterior do Amazon Linux 2. Se especificado, pelo menos um tempo de execução deve ser incluído nessa seção. Especifique um tempo de execução usando uma versão específica, uma versão principal seguida de `.x` para especificar que CodeBuild usa essa versão principal com sua versão secundária mais recente ou `latest` para usar a versão principal e secundária mais recente (por exemplo, `ruby: 3.2`, `nodejs: 18.x`, ou `java: latest`). Você pode especificar o tempo de execução usando um número ou uma variável de ambiente. Por exemplo, se você usar a imagem padrão 4.0 do Amazon Linux 2, o seguinte especificará que a versão 17 do Java, a versão secundária mais recente do python versão 3 e uma versão contida em uma variável de ambiente do Ruby sejam instaladas. Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#).

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

Você pode especificar um ou mais runtimes na seção `runtime-versions` do arquivo `buildspec`. Se o runtime depender de outro runtime, você também poderá especificar seu runtime dependente no arquivo `buildspec`. Se você não especificar nenhum tempo de

execução no arquivo buildspec, CodeBuild escolhe os tempos de execução padrão que estão disponíveis na imagem que você usa. Se você especificar um ou mais tempos de execução, CodeBuild usará somente esses tempos de execução. Se um tempo de execução dependente não for especificado, CodeBuild tentará escolher o tempo de execução dependente para você.

Se dois tempos de execução especificados entrarem em conflito, ocorrerá uma falha na compilação. Por exemplo, há um conflito entre `android: 29` e `java: openjdk11`, portanto, se ambos forem especificados, ocorrerá uma falha na compilação.

Para obter mais informações sobre runtimes disponíveis, consulte [Runtimes disponíveis](#).

 Note

Se você especificar uma `runtime-versions` seção e usar uma imagem diferente do Ubuntu Standard Image 2.0 ou posterior, ou da imagem padrão do Amazon Linux 2 (AL2) 1.0 ou posterior, a compilação emitirá o aviso "Skipping install of runtimes. Runtime version selection is not supported by this build image."

`phases/install/commands`

Sequência opcional. Contém uma sequência de escalares, em que cada escalar representa um único comando CodeBuild executado durante a instalação. CodeBuild executa cada comando, um por vez, na ordem listada, do início ao fim.

`phases/pre_build`

Sequência opcional. Representa os comandos, se houver, que são CodeBuild executados antes da compilação. Por exemplo, você poderia usar essa fase para fazer login no Amazon ECR ou instalar dependências npm.

`phases/pre_build/commands`

Sequência necessária, se `pre_build` for especificado. Contém uma sequência de escalares, em que cada escalar representa um único comando CodeBuild executado antes da compilação. CodeBuild executa cada comando, um por vez, na ordem listada, do início ao fim.

phases/build

Sequência opcional. Representa os comandos, se houver, que são CodeBuild executados durante a compilação. Por exemplo, você pode usar essa fase para executar o Mocha ou o RSpec sbt.

phases/build/commands

Necessário, se `build` for especificado. Contém uma sequência de escalares, em que cada escalar representa um único comando CodeBuild executado durante a construção. CodeBuild executa cada comando, um por vez, na ordem listada, do início ao fim.

phases/post_build

Sequência opcional. Representa os comandos, se houver, que são CodeBuild executados após a compilação. Por exemplo, é possível usar o Maven para empacotar os artefatos de compilação em um arquivo JAR ou WAR, ou enviar uma imagem do Docker ao Amazon ECR. Depois, é possível enviar uma notificação de compilação pelo Amazon SNS.

phases/post_build/commands

Necessário, se `post_build` for especificado. Contém uma sequência de escalares, em que cada escalar representa um único comando CodeBuild executado após a compilação. CodeBuild executa cada comando, um por vez, na ordem listada, do início ao fim.

relatórios

report-group-name-or-celeiro

Sequência opcional. Especifica o grupo de relatórios para o qual os relatórios são enviados. Um projeto pode ter, no máximo, cinco grupos de relatórios. Especifique o ARN de um grupo de relatórios existente ou o nome de um novo grupo de relatórios. Se você especificar um nome, CodeBuild cria um grupo de relatórios usando o nome do seu projeto e o nome especificado no formato `<project-name>-<report-group-name>`. O nome do grupo de relatórios também pode ser definido usando uma variável de ambiente no buildspec, como `$REPORT_GROUP_NAME`. Para obter mais informações, consulte [Nomenclatura do grupo de relatórios](#).

reports/<report-group>/files

Sequência necessária. Representa os locais que contêm os dados brutos dos resultados do teste gerados pelo relatório. Contém uma sequência de escalares, com cada escalar representando

um local separado onde CodeBuild pode encontrar arquivos de teste, em relação ao local de construção original ou, se definido, o. `base-directory` Os locais podem ser os seguintes:

- Um único arquivo (por exemplo, `my-test-report-file.json`).
- Um único arquivo em um subdiretório (por exemplo, `my-subdirectory/my-test-report-file.json` ou `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/*'` representa todos os arquivos recursivamente.
- `my-subdirectory/*` representa todos os arquivos em um subdiretório chamado `my-subdirectory`.
- `my-subdirectory/**/*` representa todos os arquivos recursivamente a partir de um subdiretório chamado. `my-subdirectory`

`reports/<report-group>/file-format`

Mapeamento opcional. Representa o formato do arquivo do relatório. Se não especificado, JUNITXML será usado. Esse valor não diferencia letras maiúsculas de minúsculas. Os valores possíveis são:

Relatórios de teste

CUCUMBERJSON

Cucumber JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

NUnit 3 XML

TESTNGXML

TestNG XML

VISUALSTUDIOTRX

Visual Studio TRX

Relatórios de cobertura de código

CLOVERXML

Clover XML

COBERTURAXML

Cobertura XML

JACOCOXML

JaCoCo XML

SIMPLECOV

SimpleCov JSON

Note

CodeBuild [aceita relatórios de cobertura de código JSON gerados pelo simplecov, não pelo simplecov-json.](#)

reports/<report-group>/base-directory

Mapeamento opcional. Representa um ou mais diretórios de nível superior, em relação ao local de compilação original, CodeBuild usados para determinar onde encontrar os arquivos de teste brutos.

reports/<report-group>/discard-paths

Opcional. Especifica se os diretórios de arquivos de relatório são nivelados na saída. Se isso não for especificado, ou contiver no, os arquivos de relatório serão exibidos com sua estrutura de diretório intacta. Se contiver yes, todos os arquivos de teste serão colocados no mesmo diretório de saída. Por exemplo, se um caminho para um resultado de teste for com/myapp/mytests/TestResult.xml, especificar yes colocará esse arquivo em /TestResult.xml.

Artefatos

Sequência opcional. Representa informações sobre onde é CodeBuild possível encontrar a saída da compilação e como ela é CodeBuild preparada para ser carregada no bucket de saída do S3. Essa

sequência não será necessária se, por exemplo, você estiver compilando e enviando uma imagem do Docker ao Amazon ECR, ou se estiver executando testes de unidade no código-fonte, porém não o compilando.

Note

Os metadados do Amazon S3 têm um CodeBuild cabeçalho chamado `x-amz-meta-codebuild-buildarn` que contém o `buildArn` da CodeBuild compilação que publica artefatos no Amazon S3. O `buildArn` é adicionado para permitir o rastreamento da fonte para notificações e para referenciar de qual compilação o artefato é gerado.

artifacts/files

Sequência necessária. Representa os locais que contêm os artefatos de saída da compilação, no ambiente de compilação. Contém uma sequência de escalares, com cada escalar representando um local à parte onde o CodeBuild pode encontrar artefatos de saída de compilação, relativos ao local de compilação original ou, caso definido, o diretório base. Os locais podem ser os seguintes:

- Um único arquivo (por exemplo, `my-file.jar`).
- Um único arquivo em um subdiretório (por exemplo, `my-subdirectory/my-file.jar` ou `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` representa todos os arquivos recursivamente.
- `my-subdirectory/*` representa todos os arquivos em um subdiretório chamado `my-subdirectory`.
- `my-subdirectory/**/*` representa todos os arquivos recursivamente a partir de um subdiretório chamado `my-subdirectory`.

Quando você especifica os locais dos artefatos de saída de compilação, CodeBuild pode localizar o local de construção original no ambiente de construção. Não é preciso preceder os locais de saída do artefato de build com o caminho do local de build original, ou especificar `./` ou similar. Se quiser saber o caminho para esse local, você pode executar um comando como `echo $CODEBUILD_SRC_DIR`, durante um build. O local para cada ambiente de build poderia ser um pouco diferente.

artifacts/name

Nome opcional. Especifica um nome para o artefato de compilação. Esse nome é usado quando ocorre uma das seguintes situações.

- Você usa a CodeBuild API para criar suas compilações e a `overrideArtifactName` sinalização é definida no `ProjectArtifacts` objeto quando um projeto é atualizado, um projeto é criado ou uma construção é iniciada.
- Você usa o CodeBuild console para criar suas compilações, um nome é especificado no arquivo `buildspec` e você seleciona Ativar controle de versão semântico ao criar ou atualizar um projeto. Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#).

É possível especificar um nome no arquivo `buildspec` que é calculado no momento da compilação. O nome especificado em um arquivo `buildspec` usa a linguagem de comandos do Shell. Por exemplo, você pode anexar uma data e hora ao nome do artefato para que ele seja sempre exclusivo. Os nomes de artefato exclusivos impedem que os artefatos sejam substituídos. Para obter mais informações, consulte [Shell command language](#).

- Este é um exemplo de um nome de artefato anexado com a data em que o artefato é criado.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

- Esse é um exemplo de nome de artefato que usa uma variável de CodeBuild ambiente. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

- Esse é um exemplo de nome de artefato que usa uma variável de CodeBuild ambiente com a data de criação do artefato anexada a ela.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: $AWS_REGION-$(date +%Y-%m-%d)
```

É possível adicionar informações de caminho ao nome para que os artefatos nomeados sejam colocados em diretórios com base no caminho no nome. Neste exemplo, artefatos de compilação são colocados na saída abaixo de builds/<build number>/my-artifacts.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: builds/$CODEBUILD_BUILD_NUMBER/my-artifacts
```

artifacts/discard-paths

Opcional. Especifica se os diretórios de artefatos de compilação são nivelados na saída. Se isso não for especificado, ou contiver no, os artefatos de compilação serão exibidos com sua estrutura de diretório intacta. Se contiver yes, todos os artefatos de compilação serão colocados no mesmo diretório de saída. Por exemplo, se um caminho para um arquivo no artefato de saída da compilação for com/mycompany/app/HelloWorld.java, especificar yes colocará esse arquivo em /HelloWorld.java.

artifacts/base-directory

Mapeamento opcional. Representa um ou mais diretórios de nível superior, em relação ao local da compilação original, CodeBuild usados para determinar quais arquivos e subdiretórios incluir no artefato de saída da compilação. Os valores válidos são:

- Um único diretório de nível superior (por exemplo, my-directory).

- `'my-directory*'` representa todos os diretórios de nível superior com nomes iniciados com `my-directory`.

Diretórios de nível superior correspondentes não estão incluídos no artefato de saída de build, somente seus arquivos e subdiretórios.

Você pode usar `files` e `discard-paths` para restringir que arquivos e subdiretórios serão incluídos. Por exemplo, para a seguinte estrutura de diretório:

```
.  
### my-build-1  
#   ### my-file-1.txt  
### my-build-2  
    ### my-file-2.txt  
    ### my-subdirectory  
        ### my-file-3.txt
```

E para a seguinte sequência `artifacts`:

```
artifacts:  
  files:  
    - '*/my-file-3.txt'  
  base-directory: my-build-2
```

Os seguintes subdiretório e arquivo seriam incluídos no artefato de saída de build:

```
.  
### my-subdirectory  
    ### my-file-3.txt
```

Enquanto para a seguinte sequência `artifacts`:

```
artifacts:  
  files:  
    - '**/*'  
  base-directory: 'my-build*'  
  discard-paths: yes
```

Os seguintes arquivos seriam incluídos no artefato de saída de build:

```
.
```

```
### my-file-1.txt
### my-file-2.txt
### my-file-3.txt
```

artifacts/exclude-paths

Mapeamento opcional. Representa um ou mais caminhos, em relação a `base-directory`, que CodeBuild serão excluídos dos artefatos de construção. O caractere de asterisco (*) corresponde a zero ou mais caracteres de um componente de nome sem cruzar limites da pasta. Um asterisco duplo (**) corresponde a zero ou mais caracteres de um componente de nome em todos os diretórios.

São exemplos de `exclude-paths`:

- Para excluir um arquivo de todos os diretórios: `"/file-name/"`
- Para excluir todas as pastas de pontos: `"/.*/"`
- Para excluir todos os arquivos de pontos: `"/.*"`

artifacts/enable-symlinks

Opcional. Caso o tipo de saída seja ZIP, especifica se os links simbólicos internos são preservados no arquivo ZIP. Se contiver `yes`, todos os links simbólicos internos na origem serão preservados no arquivo ZIP dos artefatos.

artifacts/s3-prefix

Opcional. Especifica um prefixo usado quando os artefatos são enviados a um bucket do Amazon S3 e o tipo de namespace é `BUILD_ID`. Quando usado, o caminho de saída no bucket é `<s3-prefix>/<build-id>/<name>.zip`.

artifacts/secondary-artifacts

Sequência opcional. Representa uma ou mais definições de artefato como um mapeamento entre um identificador e uma definição de artefato. Cada identificador de artefato deste bloco deve corresponder a um artefato definido no atributo `secondaryArtifacts` do seu projeto. Cada definição separada tem a mesma sintaxe que o bloco `artifacts` acima.

Note

A sequência [artifacts/files](#) é sempre necessária, mesmo quando há somente artefatos secundários definidos.

Por exemplo, se o projeto tem a seguinte estrutura:

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

O arquivo buildspec se parece ao seguinte:

```
version: 0.2

phases:
build:
  commands:
    - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      files:
        - directory/file1
      name: secondary-artifact-name-1
    artifact2:
      files:
        - directory/file2
      name: secondary-artifact-name-2
```

cache

Sequência opcional. Representa informações sobre onde é CodeBuild possível preparar os arquivos para carregar o cache em um bucket de cache do S3. Essa sequência não será necessária se o tipo de cache do projeto for No Cache.

cache/ chave

Sequência opcional. Representa a chave primária usada ao pesquisar ou restaurar um cache. CodeBuild faz uma correspondência exata para a chave primária.

Aqui está um exemplo da chave:

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

teclas de cache/fallback

Sequência opcional. Representa uma lista de chaves alternativas usadas sequencialmente quando um cache não pode ser encontrado usando a chave primária. Até cinco teclas alternativas são suportadas, e cada uma é combinada usando uma pesquisa de prefixo. Essa sequência será ignorada se a chave não for fornecida.

Aqui está um exemplo para as teclas de fallback:

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

cache/ação

Sequência opcional. Especifica a ação a ser executada no cache. Os valores válidos são:

- `restore` só restaura o cache sem salvar as atualizações.
- `save` só salva o cache sem restaurar uma versão anterior.

Se nenhum valor for fornecido, o CodeBuild padrão é executar a restauração e o salvamento.

cache/paths

Sequência necessária. Representa os locais do cache. Contém uma sequência de escalares, com cada escalar representando um local separado onde é CodeBuild possível encontrar

artefatos de saída de compilação, relativos ao local de construção original ou, se definido, ao diretório base. Os locais podem ser os seguintes:

- Um único arquivo (por exemplo, `my-file.jar`).
- Um único arquivo em um subdiretório (por exemplo, `my-subdirectory/my-file.jar` ou `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` representa todos os arquivos recursivamente.
- `my-subdirectory/*` representa todos os arquivos em um subdiretório chamado `my-subdirectory`.
- `my-subdirectory/**/*` representa todos os arquivos recursivamente a partir de um subdiretório chamado. `my-subdirectory`

Important

Como a declaração de buildspec deve corresponder ao YAML, o espaçamento na declaração de buildspec é importante. Se o número de espaços em sua declaração de buildspec for inválido, poderá haver falhas nas compilações imediatamente. É possível usar um validador YAML para testar se as declarações de buildspec são válidas conforme o YAML.

Se você usar o AWS CLI, ou o AWS SDKs para declarar um buildspec ao criar ou atualizar um projeto de compilação, o buildspec deverá ser uma única string expressa no formato YAML, junto com os espaços em branco necessários e os caracteres de escape de nova linha. Há um exemplo na próxima seção.

Se você usar os AWS CodePipeline consoles CodeBuild ou em vez de um arquivo buildspec.yml, poderá inserir comandos somente para a fase. `build` Em vez de usar a sintaxe anterior, você lista, em uma única linha, todos os comandos que deseja executar durante a fase de compilação. Para vários comandos, separe-os com `&&` (por exemplo, `mvn test && mvn package`).

Você pode usar os CodePipeline consoles CodeBuild ou em vez de um arquivo buildspec.yml para especificar os locais dos artefatos de saída da compilação no ambiente de compilação. Em vez de usar a sintaxe anterior, você lista, uma única linha, todos os locais. Para vários locais, separe-os com uma vírgula (por exemplo, `buildspec.yml, target/my-app.jar`).

Exemplo de buildspec

Eis um exemplo de arquivo buildspec.yml.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`

reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  files:
    - "**/*"
  base-directory: 'target/tests/reports'
```

```

    discard-paths: no
  reportGroupCucumberJson:
    files:
      - 'cucumber/target/cucumber-tests.xml'
    discard-paths: yes
    file-format: CUCUMBERJSON # default is JUNITXML
  artifacts:
    files:
      - target/messageUtil-1.0.jar
    discard-paths: yes
    secondary-artifacts:
      artifact1:
        files:
          - target/artifact-1.0.jar
        discard-paths: yes
      artifact2:
        files:
          - target/artifact-2.0.jar
        discard-paths: yes
  cache:
    paths:
      - '/root/.m2/**/*'

```

Aqui está um exemplo do buildspec anterior, expresso como uma única string, para uso com o AWS CLI ou o AWS SDKs

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-amd64\n\nparameter-store:\n  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n\nphases:\n\ninstall:\n  commands:\n    - echo Entered the install phase...\n    - apt-get update -y\n    - apt-get install -y maven\n  finally:\n    - echo This always runs even if the update or install command fails\n\npre_build:\n  commands:\n    - echo Entered the pre_build phase...\n    - docker login -u User -p $LOGIN_PASSWORD\n  finally:\n    - echo This always runs even if the login command fails\n\nbuild:\n  commands:\n    - echo Entered the build phase...\n    - echo Build started on `date`\n    - mvn install\n  finally:\n    - echo This always runs even if the install command fails\n\npost_build:\n  commands:\n    - echo Entered the post_build phase...\n    - echo Build completed on `date`\n\nreports:\n  reportGroupJUnitXml:\n    files:\n      - \"**/*\"\n    base-directory: 'target/tests/reports'\n    discard-paths: false\n  reportGroupCucumberJson:\n    files:\n      - 'cucumber/target/cucumber-tests.xml'\n    file-format: CUCUMBERJSON\n\nartifacts:\n  files:\n    - target/messageUtil-1.0.jar\n  discard-paths: yes\n  secondary-artifacts:\n    artifact1:\n      files:\n        - target/messageUtil-1.0.jar\n      discard-

```

```
paths: yes\n    artifact2:\n        files:\n            - target/messageUtil-1.0.jar\n    discard-paths: yes\n    cache:\n        paths:\n            - '/root/.m2/**/*'
```

Aqui está um exemplo dos comandos na build fase, para uso com os CodePipeline consoles CodeBuild ou.

```
echo Build started on `date` && mvn install
```

Nestes exemplos:

- É definida uma variável de ambiente personalizada, em texto simples, com a chave de `JAVA_HOME` e o valor de `/usr/lib/jvm/java-8-openjdk-amd64`.
- Uma variável de ambiente personalizada chamada `dockerLoginPassword` você armazenada no Amazon EC2 Systems Manager Parameter Store é referenciada posteriormente nos comandos de criação usando a chave `LOGIN_PASSWORD`.
- Não é possível alterar os nomes da fase de build. Os comandos que são executados neste exemplo são `apt-get update -y` e `apt-get install -y maven` (para instalar o Apache Maven), `mvn install` (para compilar, testar e empacotar o código-fonte em um artefato de saída de compilação e instalar o artefato de saída de compilação em seu repositório interno), `docker login` (para entrar no Docker com a senha que corresponde ao valor da variável de ambiente personalizada que você definiu no `dockerLoginPassword` Amazon EC2 Systems Manager Parameter Store) e vários comandos. `echo` Os `echo` comandos estão incluídos aqui para mostrar como os comandos são CodeBuild executados e a ordem em que são executados.
- `files` representa os arquivos para upload para o local de saída do build. Neste exemplo, CodeBuild carrega o arquivo `messageUtil-1.0.jar` único. O arquivo `messageUtil-1.0.jar` pode ser encontrado no diretório relativo denominado `target`, no ambiente de build. Como `discard-paths: yes` está especificado, `messageUtil-1.0.jar` é carregado diretamente (e não em um diretório `target` intermediário). O nome de arquivo `messageUtil-1.0.jar` e o nome do diretório relativo do `target` são baseados na maneira como o Apache Maven cria e armazena os artefatos de saída de build somente para este exemplo. Em seus próprios cenários, esses nomes de arquivos e diretórios serão diferentes.
- `reports` representa dois grupos de relatórios que geram relatórios durante a compilação:
 - `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` especifica o ARN de um grupo de relatórios. Os resultados do teste gerados pela estrutura de teste estão no diretório `target/tests/reports`. O formato de arquivo é `JUnitXml` e o caminho não é removido dos arquivos que contêm resultados de teste.

- `reportGroupCucumberJson` especifica um novo grupo de relatórios. Se o nome do projeto for `my-project`, um grupo de relatórios com o nome `my-project-reportGroupCucumberJson` será criado quando uma compilação for executada. Os resultados do teste gerados pela estrutura de teste estão em `cucumber/target/cucumber-tests.xml`. O formato do arquivo de teste é `CucumberJson` e o caminho é removido dos arquivos que contêm resultados de teste.

Versões de buildspec

A tabela a seguir lista as versões de buildspec e as alterações entre versões.

Versão	Alterações
0.2	• <code>environment_variables</code> foi renomeado para <code>env</code>. • <code>plaintext</code> foi renomeado para <code>variables</code>. • A propriedade <code>type</code> do <code>artifacts</code> foi desativada. • Na versão 0.1, AWS CodeBuild executa cada comando de compilação em uma instância separada do shell padrão no ambiente de compilação. Na versão 0.2, CodeBuild executa todos os comandos de compilação na mesma instância do shell padrão no ambiente de compilação.
0.1	Esta é a definição inicial do formato de especificação da compilação.

Referência de buildspec de compilação em lote

Este tópico contém a referência de buildspec das propriedades da compilação em lote.

lote

Mapeamento opcional. As configurações de compilação em lote do projeto.

batch/fast-fail

Opcional. Especifica o comportamento da compilação em lote quando uma ou mais tarefas de compilação falham.

false

O valor padrão. Todas as compilações em execução serão concluídas.

true

Todas as compilações em execução serão interrompidas quando uma das tarefas de compilação falhar.

Por padrão, todas as tarefas de compilação em lote são executadas com as configurações de compilação, como env e phases, especificadas no arquivo buildspec. É possível substituir as configurações de compilação padrão especificando valores env diferentes ou outro arquivo buildspec no parâmetro batch/*<batch-type>*/buildspec.

O conteúdo da propriedade batch varia de acordo com o tipo de compilação em lote que está sendo especificado. Os tipos possíveis de compilação em lote são:

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)
- [batch/build-fanout](#)

batch/build-graph

Define um grafo de compilação. Um grafo de compilação define um conjunto de tarefas que dependem de outras tarefas no lote. Para obter mais informações, consulte [Grafo de compilação](#).

Esse elemento contém uma matriz de tarefas de compilação. Cada etapa de compilação contém as propriedades a seguir.

Identifier

Obrigatório. O identificador da tarefa.

buildspec

Opcional. O caminho e o nome do arquivo buildspec a ser usado para essa tarefa. Se esse parâmetro não for especificado, o arquivo buildspec atual será usado.

debug-session

Opcional. Um valor booleano que indica se a depuração de sessão está habilitada para essa compilação em lote. Para obter mais informações sobre depuração de sessão, consulte [Depurar compilações com o Gerenciador de Sessões](#).

false

A depuração da sessão está desabilitada.

true

A depuração da sessão está habilitada.

depend-on

Opcional. Uma matriz de identificadores de tarefas dos quais essa tarefa depende. Essa tarefa não será executada até que essas tarefas sejam concluídas.

env

Opcional. O ambiente de compilação substitui a tarefa. Ele pode conter as seguintes propriedades:

compute-type

O identificador do tipo de computação a ser usado para a tarefa. Consulte computeType em [the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

frota

O identificador da frota a ser usada para a tarefa. Consulte [the section called “Executar compilações em frotas de capacidade reservada”](#) para obter mais informações.

image

O identificador da imagem a ser usado para a tarefa. Consulte Identificador de imagem em [the section called “Imagens do Docker fornecidas por CodeBuild”](#) para ver os valores possíveis.

privileged-mode

Valor booleano que indica se o daemon do Docker deve ser executado em um contêiner do Docker. Defina como `true` somente se o projeto de compilação for usado para criar imagens do Docker. Caso contrário, uma compilação que tente interagir com o daemon do Docker falhará. A configuração padrão é `false`.

tipo

O identificador do tipo de ambiente a ser usado para a tarefa. Consulte [Tipo de ambiente em the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

variables

As variáveis de ambiente que estarão presentes no ambiente de compilação. Consulte [env/variables](#) para obter mais informações.

Note

Observe que o tipo de computação e a frota não podem ser fornecidos no mesmo identificador de uma única compilação.

ignore-failure

Opcional. Valor booleano que indica se uma falha dessa tarefa de compilação pode ser ignorada.

false

O valor padrão. Se essa tarefa de compilação falhar, ocorrerá um erro na compilação em lote.

true

Se essa tarefa de compilação falhar, a compilação em lote ainda poderá ser bem-sucedida.

Veja um exemplo de entrada de buildspec do grafo de compilação:

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
```

```
    BUILD_ID: build1
  ignore-failure: false
- identifier: build2
  buildspec: build2.yml
  env:
    variables:
      BUILD_ID: build2
  depend-on:
    - build1
- identifier: build3
  env:
    variables:
      BUILD_ID: build3
  depend-on:
    - build2
- identifier: build4
  env:
    compute-type: ARM_LAMBDA_1GB
- identifier: build5
  env:
    fleet: fleet_name
```

batch/build-list

Define uma lista de compilações. Uma lista de compilação é usada para definir várias tarefas que são executadas paralelamente. Para obter mais informações, consulte [Lista de compilações](#).

Esse elemento contém uma matriz de tarefas de compilação. Cada etapa de compilação contém as propriedades a seguir.

Identifier

Obrigatório. O identificador da tarefa.

buildspec

Opcional. O caminho e o nome do arquivo buildspec a ser usado para essa tarefa. Se esse parâmetro não for especificado, o arquivo buildspec atual será usado.

debug-session

Opcional. Um valor booleano que indica se a depuração de sessão está habilitada para essa compilação em lote. Para obter mais informações sobre depuração de sessão, consulte [Depurar compilações com o Gerenciador de Sessões](#).

`false`

A depuração da sessão está desabilitada.

`true`

A depuração da sessão está habilitada.

`env`

Opcional. O ambiente de compilação substitui a tarefa. Ele pode conter as seguintes propriedades:

`compute-type`

O identificador do tipo de computação a ser usado para a tarefa. Consulte `computeType` em [the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

`frota`

O identificador da frota a ser usada para a tarefa. Consulte [the section called “Executar compilações em frotas de capacidade reservada”](#) para obter mais informações.

`image`

O identificador da imagem a ser usado para a tarefa. Consulte Identificador de imagem em [the section called “Imagens do Docker fornecidas por CodeBuild”](#) para ver os valores possíveis.

`privileged-mode`

Valor booleano que indica se o daemon do Docker deve ser executado em um contêiner do Docker. Defina como `true` somente se o projeto de compilação for usado para criar imagens do Docker. Caso contrário, uma compilação que tente interagir com o daemon do Docker falhará. A configuração padrão é `false`.

`tipo`

O identificador do tipo de ambiente a ser usado para a tarefa. Consulte Tipo de ambiente em [the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

`variables`

As variáveis de ambiente que estarão presentes no ambiente de compilação. Consulte [env/variables](#) para obter mais informações.

 Note

Observe que o tipo de computação e a frota não podem ser fornecidos no mesmo identificador de uma única compilação.

ignore-failure

Opcional. Valor booleano que indica se uma falha dessa tarefa de compilação pode ser ignorada.

false

O valor padrão. Se essa tarefa de compilação falhar, ocorrerá um erro na compilação em lote.

true

Se essa tarefa de compilação falhar, a compilação em lote ainda poderá ser bem-sucedida.

Veja um exemplo de entrada de buildspec da lista de compilação:

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

batch/build-matrix

Define uma matriz de compilação. Uma matriz de construção define tarefas com configurações diferentes que são executadas paralelamente. CodeBuild cria uma compilação separada para cada combinação de configuração possível. Para obter mais informações, consulte [Matriz de compilações](#).

static

As propriedades estáticas se aplicam a todas as tarefas de compilação.

ignore-failure

Opcional. Valor booleano que indica se uma falha dessa tarefa de compilação pode ser ignorada.

false

O valor padrão. Se essa tarefa de compilação falhar, ocorrerá um erro na compilação em lote.

true

Se essa tarefa de compilação falhar, a compilação em lote ainda poderá ser bem-sucedida.

env

Opcional. O ambiente de compilação substitui todas as tarefas.

privileged-mode

Valor booleano que indica se o daemon do Docker deve ser executado em um contêiner do Docker. Defina como `true` somente se o projeto de compilação for usado para criar imagens do Docker. Caso contrário, uma compilação que tente interagir com o daemon do Docker falhará. A configuração padrão é `false`.

tipo

O identificador do tipo de ambiente a ser usado para a tarefa. Consulte Tipo de ambiente em [the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

dynamic

As propriedades dinâmicas definem a matriz de compilação.

buildspec

Opcional. Matriz que contém o caminho e os nomes dos arquivos buildspec a serem usados nessas tarefas. Se esse parâmetro não for especificado, o arquivo buildspec atual será usado.

env

Opcional. O ambiente de compilação substitui essas tarefas.

compute-type

Matriz que contém os identificadores dos tipos de computação a serem usados nessas tarefas. Consulte computeType em [the section called “Modos e tipos de computação do ambiente de compilação”](#) para ver os valores possíveis.

image

Matriz que contém os identificadores das imagens a serem usados nessas tarefas.

Consulte Identificador de imagem em [the section called “Imagens do Docker fornecidas por CodeBuild”](#) para ver os valores possíveis.

variables

Matriz que contém as variáveis de ambiente que estarão presentes nos ambientes de compilação dessas tarefas. Consulte [env/variables](#) para obter mais informações.

Veja um exemplo de entrada de buildspec da matriz de compilação:

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Para obter mais informações, consulte [Matriz de compilações](#).

batch/build-fanout

Define um fanout de construção. Um fanout de compilação é usado para definir uma tarefa que é dividida em várias compilações que são executadas paralelamente. Para obter mais informações, consulte [Execute testes paralelos em compilações em lote](#).

Esse elemento contém uma tarefa de construção que pode ser dividida em várias compilações. A build-fanout seção contém as seguintes propriedades.

parallelismo

Obrigatório. O número de compilações que executarão testes em paralelo.

ignore-failure

Opcional. Um valor booleano que indica se a falha em alguma das tarefas de construção do fanout pode ser ignorada. Esse valor de ignore-failure será aplicado a todas as compilações do fanout.

false

O valor padrão. Se alguma tarefa de compilação do fanout falhar, a compilação em lote falhará.

true

Se alguma tarefa de compilação do fanout falhar, a compilação em lote ainda poderá ser bem-sucedida.

Veja a seguir um exemplo de uma entrada buildspec do build fanout:

```
version: 0.2

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

phases:
  install:
```

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

Para ter mais informações, consulte [Crie um fanout](#) e [Use o comando codebuild-tests-run CLI](#).

Crie uma referência de ambiente para AWS CodeBuild

Ao ligar AWS CodeBuild para executar uma compilação, você deve fornecer informações sobre o ambiente de compilação. Um ambiente de compilação representa uma combinação de sistema operacional, tempo de execução da linguagem de programação e ferramentas CodeBuild usadas para executar uma compilação. Para obter informações sobre como um ambiente de computação funciona, consulte [Como CodeBuild funciona](#).

Um ambiente de build contém uma imagem Docker. Para obter mais informações, consulte o [glossário do Docker](#) no site Docker Docs.

Ao fornecer informações CodeBuild sobre o ambiente de compilação, você especifica o identificador de uma imagem do Docker em um tipo de repositório compatível. Isso inclui o repositório de imagens do CodeBuild Docker, imagens publicamente disponíveis no Docker Hub e repositórios do Amazon Elastic Container Registry (Amazon ECR) que AWS sua conta tem permissão para acessar.

- Recomendamos que você use imagens do Docker armazenadas no repositório de imagens do CodeBuild Docker, pois elas são otimizadas para uso com o serviço. Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#).
- Para obter o identificador de uma imagem de Docker publicamente disponível e armazenada no Docker Hub, consulte [Searching for Repositories](#) no site do Docker Docs.
- Para saber como trabalhar com imagens do Docker armazenadas em repositórios do Amazon ECR na conta da AWS, consulte [Exemplo do Amazon ECR](#).

Além de um identificador de imagem de Docker, você também pode especificar um conjunto de recursos computacionais que o ambiente de compilação utiliza. Para obter mais informações, consulte [Modos e tipos de computação do ambiente de compilação](#).

Tópicos

- [Imagens do Docker fornecidas por CodeBuild](#)
- [Modos e tipos de computação do ambiente de compilação](#)
- [Shells e comandos em ambientes de compilação](#)
- [Variáveis de ambiente em ambientes de compilação](#)
- [Tarefas em segundo plano em ambientes de compilação](#)

Imagens do Docker fornecidas por CodeBuild

Uma imagem compatível é a versão principal mais recente de uma imagem disponível CodeBuild e é atualizada com atualizações secundárias e de patch. CodeBuild otimiza a duração do provisionamento de compilações com imagens compatíveis, armazenando-as em cache nas Amazon Machine [Images \(AMI\) da máquina](#). Se você quiser se beneficiar do armazenamento em cache e minimizar a duração do provisionamento de sua compilação, selecione Sempre usar a imagem mais recente para essa versão de tempo de execução na seção Versão da imagem do CodeBuild console, em vez de uma versão mais granular, como. `aws/codebuild/amazonlinux-x86_64-standard:4.0-1.0.0`

Tópicos

- [Obter a lista das imagens do Docker atuais](#)
- [EC2 imagens de computação](#)
- [Imagens de computação do Lambda](#)
- [Imagens obsoletas CodeBuild](#)
- [Runtimes disponíveis](#)
- [Versões do tempo de execução](#)

Obter a lista das imagens do Docker atuais

CodeBuild atualiza frequentemente a lista de imagens do Docker para adicionar as imagens mais recentes e descontinuar as imagens antigas. Para obter a lista mais atual, faça um dos seguintes procedimentos:

- No CodeBuild console, no assistente Criar projeto de compilação ou na página Editar projeto de compilação, em Imagem do ambiente, escolha Imagem gerenciada. Selecione nas listas suspensas Operating system (Sistema operacional), Runtime (Tempo de execução) e Runtime version (Versão do tempo de execução). Para ter mais informações, consulte [Criar um projeto de compilação \(console\)](#) ou [Alterar configurações de um projeto de compilação \(console\)](#).
- Para o AWS CLI, execute o `list-curated-environment-images` comando:

```
aws codebuild list-curated-environment-images
```


- Para o AWS SDKs, chame a `ListCuratedEnvironmentImages` operação da sua linguagem de programação de destino. Para obter mais informações, consulte o [AWS SDKs e referência de ferramentas](#).

EC2 imagens de computação

AWS CodeBuild suporta as seguintes imagens do Docker que estão disponíveis para EC2 computação em. CodeBuild

Note

A imagem base da plataforma Windows Server Core 2019 só está disponível nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (Oregon)
- Europa (Irlanda)

Plataforma	Identificador da imagem	Definição
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/padrão/4.0
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/padrão/5.0
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto8	al/standard/corretto8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto11	al/standard/corretto11

Plataforma	Identificador da imagem	Definição
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D
Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
macOS	aws/codebuild/macos-arm-base:14	N/D

Note

Em 22 de novembro de 2024, os aliases para imagens de tempo de execução padrão baseadas em Linux foram atualizados de para. amazonlinux2 amazonlinux Nenhuma atualização manual é necessária, pois os aliases anteriores ainda são válidos.

Imagens de computação do Lambda

AWS CodeBuild suporta as seguintes imagens do Docker que estão disponíveis para AWS Lambda computação em. CodeBuild

aarch64Arquitetura do

Plataforma	Identificador da imagem	Definição
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6	al- lambda/aarch64/dotnet 6
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8	todos os lambda/aarch64/dotnet 8
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21	tudo- lambda/aarch64/go 1,21
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.24	al- lambda/aarch64/go 1,24
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11	tudo- lambda/aarch64/corretto 11

Plataforma	Identificador da imagem	Definição
	mbda-standard:corretto11	
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17	tudo- lambda/aarch64/corretto17
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21	tudo- lambda/aarch64/corretto21
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18	tudo- lambda/aarch64/nodejs18
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20	al- lambda/aarch64/nodejs 20
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs22	tudo- lambda/aarch64/nodejs22
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11	al- lambda/aarch64/python3.11

Plataforma	Identificador da imagem	Definição
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12	al-lambda/aarch64/python3,12
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.13	al-lambda/aarch64/python3,13
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2	al-lambda/aarch64/ruby3,2
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.4	al-lambda/aarch64/ruby3.4

x86_64Arquitetura do

Plataforma	Identificador da imagem	Definição
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6	al-lambda/x86_64/dotnet6
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8	todos os lambda/x86_64/dotnet8

Plataforma	Identificador da imagem	Definição
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21	tudo- lambda/x86_64/go 1,21
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.24	al- lambda/x86_64/go 1,24
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11	tudo- lambda/x86_64/corretto 11
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17	tudo- lambda/x86_64/corretto 17
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21	tudo- lambda/x86_64/corretto 21
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18	tudo- lambda/x86_64/nodejs 18
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20	al- lambda/x86_64/nodejs 20

Plataforma	Identificador da imagem	Definição
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs22	al-lambda/x86_64/nodejs22
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11	al-lambda/x86_64/python 3.11
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12	al-lambda/x86_64/python 3,12
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.13	al-lambda/x86_64/python 3,13
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2	al-lambda/x86_64/ruby 3,2
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.4	al-lambda/x86_64/ruby 3.4

Imagens obsoletas CodeBuild

Uma imagem obsoleta é uma imagem que não é mais armazenada em cache nem atualizada pelo. CodeBuild Uma imagem obsoleta não recebe mais atualizações de versão secundária nem atualizações de versão de patch e, como não estão mais atualizadas, usá-las pode não ser seguro.

Se seu CodeBuild projeto estiver configurado para usar uma versão de imagem mais antiga, o processo de provisionamento baixará essa imagem do docker e a usará para criar o ambiente de tempo de execução em contêineres, o que pode aumentar a duração do provisionamento e a duração geral da compilação.

CodeBuild descontinuou as seguintes imagens do Docker. Ainda é possível usar essas imagens, mas elas não serão armazenadas em cache no host de compilação e resultarão em tempos de provisionamento mais altos.

Plataforma	Identificador da imagem	Definição	Data da substituição
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:3.0	al2/standard/3.0	9 de maio de 2023
Ubuntu 18.04	aws/codebuild/ standard:4.0	ubuntu/standard/4.0	31 de março de 2023
Amazon Linux 2	aws/codebuild/ amazonlinux2- aarch64-s tandard:1.0	al2/aarch64/standa rd/1,0	31 de março de 2023
Ubuntu 18.04	aws/codebuild/ standard:3.0	ubuntu/standard/3.0	30 de junho de 2022
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:2.0	al2/standard/2.0	30 de junho de 2022

Tópicos

- [Runtimes disponíveis](#)
- [Versões do tempo de execução](#)

Runtimes disponíveis

Você pode especificar um ou mais runtimes na seção `runtime-versions` do arquivo `buildspec`. Se o runtime depender de outro runtime, você também poderá especificar seu runtime dependente no arquivo `buildspec`. Se você não especificar nenhum tempo de execução no arquivo `buildspec`, CodeBuild escolhe os tempos de execução padrão que estão disponíveis na imagem que você usa. Se você especificar um ou mais tempos de execução, CodeBuild usará somente esses tempos de execução. Se um tempo de execução dependente não for especificado, CodeBuild tentará escolher o tempo de execução dependente para você. Para obter mais informações, consulte [Specify runtime versions in the buildspec file](#).

Tópicos

- [Runtimes de imagens do Linux](#)
- [Runtimes de imagens do macOS](#)
- [Runtimes de imagens do Windows](#)

Runtimes de imagens do Linux

A tabela a seguir contém os runtimes disponíveis e as imagens padrão do Linux compatíveis.

Runtimes das plataformas Ubuntu e Amazon Linux

Nome do runtime	Versão	Imagens
dotnet	3.1	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	5.0	Padrão do Ubuntu: 5.0
	6.0	Padrão do Amazon Linux 2 x86_64 Lambda: dotnet6 Padrão Amazon Linux 2 AArch64 Lambda: dotnet6

Nome do runtime	Versão	Imagens
		Padrão do Amazon Linux 2 x86_64: 4.0
		Padrão do Amazon Linux 2023 x86_64: 5.0
		AArch64Padrão Amazon Linux 2023:3.0
		Padrão do Ubuntu: 6.0
		Padrão do Ubuntu: 7.0
	8.0	Padrão do Amazon Linux 2023 x86_64: 5.0
		AArch64Padrão Amazon Linux 2023:3.0
		Padrão do Ubuntu: 7.0
golang	1.12	AArch64Padrão Amazon Linux 2:2.0
	1.13	AArch64Padrão Amazon Linux 2:2.0
	1.14	AArch64Padrão Amazon Linux 2:2.0
	1.15	Padrão do Ubuntu: 5.0
	1.16	Padrão do Ubuntu: 5.0

Nome do runtime	Versão	Imagens
	1,18	Padrão do Amazon Linux 2 x86_64: 4.0 Padrão do Ubuntu: 6.0
	1,20	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	1,21	Padrão do Amazon Linux 2 x86_64 Lambda: go1.21 Padrão Amazon Linux 2 AArch64 Lambda: go1.21 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	1,22	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	1,23	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	1,24	Amazon Linux 2023 x86_64 Lambda padrão: go1.24 Padrão Amazon Linux 2023 AArch64 Lambda: go1.24
java	corretto8	Padrão do Amazon Linux 2 x86_64: corretto8 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2:2.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 5.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	corretto11	Padrão do Amazon Linux 2 x86_64: corretto11
		Padrão do Amazon Linux 2 x86_64 Lambda: corretto11
		Padrão do Amazon Linux 2023 x86_64: 5.0
		Padrão Amazon Linux 2 AArch64 Lambda: corretto11
		AArch64Padrão Amazon Linux 2:2.0
		AArch64Padrão Amazon Linux 2023:3.0
		Padrão do Ubuntu: 5.0
		Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	corretto17	Padrão do Amazon Linux 2 x86_64 Lambda: corretto17 Padrão Amazon Linux 2 AArch64 Lambda: corretto17 Padrão do Amazon Linux 2 x86_64: 4.0 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 6.0 Padrão do Ubuntu: 7.0
	corretto21	Padrão do Amazon Linux 2 x86_64 Lambda: corretto21 Padrão Amazon Linux 2 AArch64 Lambda: corretto21 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
nodejs	10	AArch64Padrão Amazon Linux 2:2.0
	12	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	14	Padrão do Ubuntu: 5.0
	16	Padrão do Amazon Linux 2 x86_64: 4.0 Padrão do Ubuntu: 6.0
	18	Padrão do Amazon Linux 2 x86_64 Lambda: nodejs18 Padrão Amazon Linux 2 AArch64 Lambda: nodejs18 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	20	Padrão do Amazon Linux 2 x86_64 Lambda: nodejs20 Padrão Amazon Linux 2 AArch64 Lambda: nodejs20 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	22	Amazon Linux 2023 x86_64 Lambda padrão: nodejs22 Padrão Amazon Linux 2023 AArch64 Lambda: nodejs22 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
php	7.3	AArch64 Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0

Nome do runtime	Versão	Imagens
	7.4	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	8.0	Padrão do Ubuntu: 5.0
	8.1	Padrão do Amazon Linux 2 x86_64: 4.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 6.0
	8.2	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	8.3	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
python	3.7	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0

Nome do runtime	Versão	Imagens
	3.8	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	3.9	Padrão do Amazon Linux 2 x86_64: 4.0 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2:2.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 5.0 Padrão do Ubuntu: 7.0
	3.10	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 6.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	3.11	Padrão do Amazon Linux 2 x86_64 Lambda: python3.11 Padrão Amazon Linux 2 AArch64 Lambda: python3.11 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	3.12	Padrão do Amazon Linux 2 x86_64 Lambda: python3.12 Padrão Amazon Linux 2 AArch64 Lambda: python3.12 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	3.13	Amazon Linux 2023 x86_64 Lambda padrão: python3.13 Padrão Amazon Linux 2023 AArch64 Lambda: python3.13 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
ruby	2.6	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	2.7	AArch64Padrão Amazon Linux 2:2.0 Padrão do Ubuntu: 5.0
	3.1	Padrão do Amazon Linux 2 x86_64: 4.0 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 6.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	3.2	Padrão do Amazon Linux 2 x86_64 Lambda: ruby3.2 Padrão Amazon Linux 2 AArch64 Lambda: ruby3.2 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0
	3.3	Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64 Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Nome do runtime	Versão	Imagens
	3.4	Amazon Linux 2023 x86_64 Lambda padrão: ruby3.4 Padrão Amazon Linux 2023 AArch64 Lambda: ruby3.4 Padrão do Amazon Linux 2023 x86_64: 5.0 AArch64Padrão Amazon Linux 2023:3.0 Padrão do Ubuntu: 7.0

Runtimes de imagens do macOS

Important

As imagens CodeBuild selecionadas para compilações do Mac contêm o macOS e o Xcode pré-instalados. Ao usar o software Xcode, você reconhece, compreende e concorda com o Contrato do [Xcode e da Apple SDKs](#). Se você não aceita os termos e condições do contrato, não use o software Xcode. Em vez disso, forneça suas próprias imagens de máquina da Amazon (AMIs). Para obter mais informações, consulte [Como configurar uma frota macOS de capacidade reservada?](#).

A tabela a seguir contém os runtimes disponíveis compatíveis com macOS.

Runtimes da plataforma macOS

Nome do runtime	Versão	Imagens	Notas adicionais
bash	3.2.57	macos-arm-base1:14 macos-arm-base1:15	

Nome do runtime	Versão	Imagens	Notas adicionais
clang	15.0.0	macos-arm-base1:14	
	16.0.0	macos-arm-base1:15	
dotnet sdk	8.0.406	macos-arm-base1:14	
		macos-arm-base1:15	
gcc	11.5.0	macos-arm-base1:14	Disponível usando o alias gcc-11
		macos-arm-base1:15	
	12.4.0	macos-arm-base1:14	Disponível usando o alias gcc-12
		macos-arm-base1:15	
	13.3.0	macos-arm-base1:14	Disponível usando o alias gcc-13
		macos-arm-base1:15	
	14.2.0	macos-arm-base1:14	Disponível usando o alias gcc-14
		macos-arm-base1:15	
gnu	11.5.0	macos-arm-base1:14	Disponível usando o alias gfortran-11
		macos-arm-base1:15	
	12.4.0	macos-arm-base1:14	Disponível usando o alias gfortran-12
		macos-arm-base1:15	
	13.3.0	macos-arm-base1:14	Disponível usando o alias gfortran-13
		macos-arm-base1:15	
	14.2.0	macos-arm-base1:14	Disponível usando o alias gfortran-14
		macos-arm-base1:15	

Nome do runtime	Versão	Imagens	Notas adicionais
golang	1.22.12	macos-arm-base1:14	
		macos-arm-base1:15	
	1.23.6	macos-arm-base1:14	
		macos-arm-base1:15	
	1.24.0	macos-arm-base1:14	
		macos-arm-base1:15	
java	Corretto8	macos-arm-base1:14	
		macos-arm-base1:15	
	Corretto11	macos-arm-base1:14	
		macos-arm-base1:15	
	Corretto17	macos-arm-base1:14	
		macos-arm-base1:15	
	Corretto21	macos-arm-base1:14	
		macos-arm-base1:15	
kotlin	2.1.10	macos-arm-base1:14	
		macos-arm-base1:15	
mono	6.12.0	macos-arm-base1:14	
		macos-arm-base1:15	
nodejs	18.20.7	macos-arm-base1:14	

Nome do runtime	Versão	Imagens	Notas adicionais
	20.18.3	macos-arm-base1:14	
		macos-arm-base1:15	
	22.14.0	macos-arm-base1:14	
		macos-arm-base1:15	
perl	5.34.1	macos-arm-base1:14	
		macos-arm-base1:15	
php	8.1.31	macos-arm-base1:14	
	8.2.27	macos-arm-base1:14	
		macos-arm-base1:15	
	8.3.17	macos-arm-base1:14	
		macos-arm-base1:15	
	8.4.4	macos-arm-base1:14	
		macos-arm-base1:15	
python	3.9.21	macos-arm-base1:14	
	3.10.16	macos-arm-base1:14	
		macos-arm-base1:15	
	3.11.11	macos-arm-base1:14	
		macos-arm-base1:15	
	3.12.9	macos-arm-base1:14	
		macos-arm-base1:15	

Nome do runtime	Versão	Imagens	Notas adicionais
ruby	3.13.2	macos-arm-base1:14	
		macos-arm-base1:15	
	3.1.6	macos-arm-base1:14	
	3.2.7	macos-arm-base1:14	
		macos-arm-base1:15	
	3.3.7	macos-arm-base1:14	
		macos-arm-base1:15	
	3.4.2	macos-arm-base1:14	
		macos-arm-base1:15	
rust	1.85.0	macos-arm-base1:14	
		macos-arm-base1:15	
swift	5.10.0.13	macos-arm-base1:14	
	6.0.3.1.10	macos-arm-base1:14	
Xcode	15.4	macos-arm-base1:14	
	16.2	macos-arm-base1:15	

Runtimes de imagens do Windows

A imagem base do Windows Server Core 2019 contém os runtimes a seguir.

Runtimes da plataforma Windows

Nome do runtime	Windows Server Core 2019 versões 1.0	Windows Server Core 2019 versões 2.0	Windows Server Core 2019 versões 3.0
dotnet	3.1	3.1	8.0

Nome do runtime	Windows Server Core 2019 versões 1.0	Windows Server Core 2019 versões 2.0	Windows Server Core 2019 versões 3.0
	5,0	6.0	
		7.0	
dotnet sdk	3.1	3.1	8.0
	5,0	6.0	
		7.0	
golang	1.14	1,18	1,21
			1,22
			1,23
gradle	6.7	7.6	8.12
java	Corretto11	Corretto11	Corretto8
		Corretto17	Corretto11
			Corretto17
			Corretto21
maven	3.6	3.8	3.9
nodejs	14.15	16.19	20,18
			22.13
php	7.4	8.1	8.3
			8.4
powershell	7.1	7.2	7.4

Nome do runtime	Windows Server Core 2019 standard versões 1.0	Windows Server Core 2019 standard versões 2.0	Windows Server Core 2019 standard versões 3.0
python	3.8	3.10	3.10 3.11 3.12 3.13
ruby	2.7	3.1	3.2 3.3 3.4

Versões do tempo de execução

Ao especificar um tempo de execução na seção [runtime-versions](#) do arquivo buildspec, é possível indicar uma versão específica, uma versão principal específica e a versão secundária mais recente, ou a versão mais recente. A tabela a seguir lista os tempos de execução disponíveis e como especificá-los. Nem todas as versões de runtime estão disponíveis em todas as imagens. A seleção da versão de runtime também não é compatível com as imagens personalizadas. Para obter mais informações, consulte [Runtimes disponíveis](#). Se você quiser instalar e usar uma versão de runtime personalizada em vez das versões de tempo de execução pré-instaladas, consulte [Versões de runtime personalizadas](#).

Versões de runtime das plataformas Ubuntu e Amazon Linux 2

Nome do runtime	Versão	Versão específica	Versão secundária mais recente e principal específica	Versão mais recente
android	28	android: 28	android: 28.x	android: latest
	29	android: 29	android: 29.x	

Nome do runtime	Versão	Versão específica	Versão secundária mais recente e principal específica	Versão mais recente
dotnet	3.1	dotnet: 3.1	dotnet: 3.x	dotnet: latest
	5,0	dotnet: 5.0	dotnet: 5.x	
	6.0	dotnet: 6.0	dotnet: 6.x	
	8.0	dotnet: 8.0	dotnet: 8.x	
golang	1.12	golang: 1.12	golang: 1.x	golang: latest
	1.13	golang: 1.13		
	1.14	golang: 1.14		
	1.15	golang: 1.15		
	1.16	golang: 1.16		
	1,18	golang: 1.18		
	1,20	golang: 1.20		
	1,21	golang: 1.21		
	1,22	golang: 1.22		
	1,23	golang: 1.23		
java	corretto8	java: corretto	java: corretto .x	java: latest
	corretto11	java: corretto 1	java: corretto 1.x	

Nome do runtime	Versão	Versão específica	Versão secundária mais recente e principal específica	Versão mais recente
	corretto17	java: corretto 7	java: corretto 7.x	
	corretto21	java: corretto 1	java: corretto 1.x	
nodejs	10	nodejs: 10	nodejs: 10.x	nodejs: latest
	12	nodejs: 12	nodejs: 12.x	
	14	nodejs: 14	nodejs: 14.x	
	16	nodejs: 16	nodejs: 16.x	
	18	nodejs: 18	nodejs: 18.x	
	20	nodejs: 20	nodejs: 20.x	
	22	nodejs: 22	nodejs: 22.x	
php	7.3	php: 7.3	php: 7.x	php: latest
	7.4	php: 7.4	php: 8.x	
	8.0	php: 8.0		
	8.1	php: 8.1		
	8.2	php: 8.2		
	8.3	php: 8.3		
python	3.7	python: 3.7	python: 3.x	python: latest
	3.8	python: 3.8		

Nome do runtime	Versão	Versão específica	Versão secundária mais recente e principal específica	Versão mais recente
	3.9	python: 3.9		
	3.10	python: 3.10		
	3.11	python: 3.11		
	3.12	python: 3.12		
	3.13	python: 3.13		
ruby	2.6	ruby: 2.6	ruby: 2.x	ruby: latest
	2.7	ruby: 2.7	ruby: 3.x	
	3.1	ruby: 3.1		
	3.2	ruby: 3.2		
	3.3	ruby: 3.3		
	3.4	ruby: 3.4		

Você pode usar uma especificação de compilação para instalar outros componentes (por exemplo AWS CLI, Apache Maven, Apache Ant, Mocha ou similares) durante a fase de compilação. RSpec `install` Para obter mais informações, consulte [Exemplo de buildspec](#).

Versões de runtime personalizadas

Em vez de usar as versões de tempo de execução CodeBuild pré-instaladas em imagens gerenciadas, você pode instalar e usar as versões personalizadas de sua escolha. A tabela a seguir lista os runtimes personalizados disponíveis e como especificá-los.

Note

A seleção da versão de runtime personalizada só é compatível com imagens do Ubuntu e Amazon Linux.

Versões de runtime personalizadas

Nome do runtime	Sintaxe	Exemplo
dotnet	<i><major>.<minor>.<patch></i>	5.0.408
golang	<i><major>.<minor></i>	1.19
	<i><major>.<minor>.<patch></i>	1.19.1
java	corretto <i><major></i>	corretto15
nodejs	<i><major></i>	14
	<i><major>.<minor></i>	14.21
	<i><major>.<minor>.<patch></i>	14.21.3
php	<i><major>.<minor>.<patch></i>	8.0.30
python	<i><major></i>	3
	<i><major>.<minor></i>	3.7
	<i><major>.<minor>.<patch></i>	3.7.16
ruby	<i><major>.<minor>.<patch></i>	3.0.6

Exemplo de buildspec de runtime personalizado

Veja um exemplo de buildspec que especifica versões de runtime personalizadas.

```
version: 0.2
phases:
  install:
```



```
runtime-versions:
  java: corretto15
  php: 8.0.30
  ruby: 3.0.6
  golang: 1.19
  python: 3.7
  nodejs: 14
  dotnet: 5.0.408
```

Modos e tipos de computação do ambiente de compilação

Em CodeBuild, você pode especificar a imagem do ambiente de computação e tempo de execução que CodeBuild usa para executar suas compilações. Computação se refere ao mecanismo de computação (CPU, memória e sistema operacional) que é gerenciado e mantido pela CodeBuild. Uma imagem de ambiente de runtime é uma imagem de contêiner executada na plataforma de computação escolhida e inclui ferramentas extras de que sua compilação pode precisar, como a AWS CLI.

Tópicos

- [Sobre computação](#)
- [Sobre os tipos de ambiente de capacidade reservada](#)
- [Sobre os tipos de ambiente sob demanda](#)

Sobre computação

CodeBuild ofertas EC2 e modos de AWS Lambda computação. EC2 oferece flexibilidade otimizada durante sua construção e AWS Lambda oferece velocidades de inicialização otimizadas. AWS Lambda suporta compilações mais rápidas devido a uma menor latência de inicialização. AWS Lambda também é dimensionado automaticamente, para que as compilações não fiquem esperando na fila para serem executadas. Para obter mais informações, consulte [Execute compilações em computação AWS Lambda](#).

Com o modo de EC2 computação, você pode executar suas compilações com frotas de capacidade reservada ou sob demanda. Para frotas sob demanda, você pode selecionar tipos de computação predefinidos, como ou. BUILD_GENERAL1_SMALL BUILD_GENERAL1_LARGE Para obter mais informações, consulte [Sobre os tipos de ambiente sob demanda](#). Para frotas de capacidade reservada, você pode selecionar suas configurações de computação, incluindo vCPU, memória e espaço em disco. Depois de especificar as configurações, CodeBuild escolherá um tipo de

computação compatível que atenda aos seus requisitos. Para obter mais informações, consulte [Sobre os tipos de ambiente de capacidade reservada](#).

Sobre os tipos de ambiente de capacidade reservada

AWS CodeBuild fornece tipos de ambiente Linux x86, Arm, GPU, Windows e macOS para frotas de capacidade reservada. A tabela a seguir mostra o tipo de máquina, a memória CPUs, v e o espaço em disco disponíveis classificados por região:

US East (N. Virginia)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu. .128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
GPU Linux	96	384 GiB	3785 GB (SSD)	NVME	reserved.gpu.96cpu.384gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

US East (Ohio)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gi b.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvm e

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

US West (Oregon)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
GPU Linux	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
macOS	12	32 GiB	256 GB	GENERAL	reserved. arm.m2.12 cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Tokyo)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Mumbai)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gi b.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvm e

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Singapore)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Sydney)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	16	32 GiB	256 GB	GENERAL	reserved. arm.16cpu .32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved. arm.32cpu .64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved. arm.48cpu .96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu .128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Europe (Frankfurt)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gi b.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvm e

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Europe (Ireland)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
Arm	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved. arm.16cpu .32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved. arm.32cpu .64gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
Arm	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

South America (São Paulo)

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Arm	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
Arm	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Arm	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
Arm	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
Arm	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
BRAÇO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRAÇO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRAÇO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipo de ambiente	v CPUs	Memória	Espaço em disco	Tipo de máquina	Tipo de instância de computação
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Janelas EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Janelas EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Para obter mais informações sobre o identificador de preços, consulte <https://aws.amazon.com/codebuild/pricing/>.

Para escolher um tipo de computação:

- No CodeBuild console, na página de configuração da frota de computação, escolha uma das opções entre v CPUs, Memória e Disco. Para obter mais informações, consulte [Criar uma frota de capacidade reservada](#).

- Para o AWS CLI, execute o `update-fleet` comando `create-fleet` or, especificando o valor de `computeType` to `ATTRIBUTE_BASED_COMPUTE`. [Para obter mais informações, consulte `create-fleet` ou `update-fleet`.](#)
- Para o AWS SDKs, chame o equivalente da `UpdateFleet` operação `CreateFleet` ou para sua linguagem de programação de destino, especificando o valor de `computeType` to `ATTRIBUTE_BASED_COMPUTE`. Para obter mais informações, consulte o [AWS SDKs e referência de ferramentas](#).

Note

Para o AWS CLI e AWS SDKs, você ainda pode usar `computeType` entradas como `BUILD_GENERAL1_SMALL`, para escolher os tipos de computação em vez de. `ATTRIBUTE_BASED_COMPUTE` Para obter mais informações, consulte [Sobre os tipos de ambiente sob demanda](#).

Famílias de instâncias com suporte

AWS CodeBuild oferece suporte às seguintes instâncias para frotas de capacidade reservada. :

- Uso geral: M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8g | T3 | T3a | T4g
- Otimizadas para computação: C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8g
- Memória otimizada: R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7gd | R7i | R7iz | R8g | U-3tb1 | U-6tb1 | U-6tb1 | U-6tb1 | U-9TB1 | U-12Tb1 | U-18TB1 | U-24Tb1 | U7i-6Tb | U7i-8Tb | U7i-12Tb | U7in-16TB | U7in-24TB | U7i-32TB | X1 | X1E | X2gD | X2iEDN | X2iEzN | X8G | z1d
- Armazenamento otimizado: D3 | D3en | I3 | I3en | i4G | i4i | i7le | i8G | i4GN | IS4gen
- Computação acelerada: DL1 | | F1 DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | Gr6 | Inf1 | Inf2 | P3 | P3dn | P4d | P5 | P5e | P5en | Trn1 | Trn1n | Trn2 | Trn2 | VT1
- Com computação de alta performance: Hpc6a | Hpc6id | Hpc7a | Hpc7g
- Geração anterior: A1

Para criar uma frota de capacidade reservada com um tipo de instância específico:

- No CodeBuild console, na página de configuração da frota de computação, navegue até a seção Configuração de capacidade. No modo de seleção de computação, selecione Entrada manual e, em Tipo de instância de computação, escolha um dos tipos de instância no menu suspenso. Para obter mais informações, consulte [Criar uma frota de capacidade reservada](#).
- Para o AWS CLI, execute o `update-fleet` comando `create-fleet` or, especificando o valor de `computeType` to `CUSTOM_INSTANCE_TYPE` e to `ComputeConfiguration instanceType` para o tipo de instância especificado. [Para obter mais informações, consulte `create-fleet` ou `update-fleet`](#).
- Para o AWS SDKs, chame o equivalente da `UpdateFleet` operação `CreateFleet` ou para sua linguagem de programação de destino, especificando o valor de `computeType` to `CUSTOM_INSTANCE_TYPE` e to `ComputeConfiguration instanceType` para o tipo de instância especificado. Para obter mais informações, consulte o [AWS SDKs e referência de ferramentas](#).

Sobre os tipos de ambiente sob demanda

AWS CodeBuild fornece ambientes de criação com a seguinte memória CPUs, v e espaço em disco disponíveis para o modo de EC2 computação:

Tipo de computação	Valor <code>computeType</code> do ambiente	Valor de tipo do ambiente	Memória	v CPUs	Espaço em disco
ARM Small ¹	BUILD_GENERAL1_SMALL	ARM_CONTAINER ARM_EC2	4 GiB	2	64 GB
ARM Medium ¹	BUILD_GENERAL1_MEDIUM	ARM_CONTAINER ARM_EC2	8 GiB	4	128 GB
ARM Large ¹	BUILD_GENERAL1_LARGE	ARM_CONTAINER ARM_EC2	16 GiB	8	128 GB

Tipo de computação	Valor computeType do ambiente	Valor de tipo do ambiente	Memória	v CPUs	Espaço em disco
BRAÇO XLarge ¹	BUILD_GENERAL1_XLARGE	ARM_CONTAINER	64 GiB	32	256 GB
ARM 2 XLarge ¹	BUILD_GENERAL1_2XLARGE	ARM_CONTAINER	96 GiB	48	824 GB
Linux Small ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER LINUX_EC2	4 GiB	2	64 GB
Linux Medium ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER LINUX_EC2	8 GiB	4	128 GB
Linux Large ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER LINUX_EC2	16 GiB	8	128 GB
Linux XLarge ¹	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	72 GiB	36	256 GB
Linux 2 XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	144 GiB	72	824 GB (SSD)
Linux GPU Small	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GiB	4	235 GB (SSD)

Tipo de computação	Valor computeType do ambiente	Valor de tipo do ambiente	Memória	v CPUs	Espaço em disco
Linux GPU Large	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GiB	32	50 GB
Windows Medium ¹	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	8 GiB	4	128 GB
		WINDOWS_SERVER_2022_CONTAINER			
		WINDOWS_EC2			
Windows Large ¹	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER	16 GiB	8	128 GB
		WINDOWS_SERVER_2022_CONTAINER			
		WINDOWS_EC2			
Windows XLarge ¹	BUILD_GENERAL1_XLARGE	WINDOWS_SERVER_2022_CONTAINER	72 GiB	36	256 GB

Tipo de computação	Valor computeType do ambiente	Valor de tipo do ambiente	Memória	v CPUs	Espaço em disco
Windows XLarge 2 ¹	BUILD_GENERAL1_2XLARGE	WINDOWS_SERVER_2022_CONTAINER	144 GiB	72	824 GB

¹ A versão mais recente desse tipo de imagem é armazenada em cache. Se você especificar uma versão mais específica, CodeBuild provisionará essa versão em vez da versão em cache. Isso pode resultar em tempos de compilação mais longos. Por exemplo, para se beneficiar do cache, especifique `aws/codebuild/amazonlinux-x86_64-standard:5.0` em vez de uma versão mais granular, como `aws/codebuild/amazonlinux-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild fornece ambientes de criação com a seguinte memória e espaço em disco disponíveis para o modo de AWS Lambda computação:

Tipo de computação	Valor computeType do ambiente	Valor de tipo do ambiente	Memória	Espaço em disco
ARM Lambda de 1 GB	BUILD_LAMBDA_1GB	ARM_LAMBDA_CONTAINER	15 GiB	10 GB
ARM Lambda de 2 GB	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GiB	10 GB
ARM Lambda de 4 GB	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GiB	10 GB
ARM Lambda de 8 GB	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GiB	10 GB

Tipo de computação	Valor computeType do ambiente	Valor de tipo do ambiente	Memória	Espaço em disco
ARM Lambda de 10 GB	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 GiB	10 GB
Linux Lambda de 1 GB	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	15 GiB	10 GB
Linux Lambda de 2 GB	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GiB	10 GB
Linux Lambda de 4 GB	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GiB	10 GB
Linux Lambda de 8 GB	BUILD_LAMBDA_8GB	LINUX_LAMBDA_CONTAINER	8 GiB	10 GB
Linux Lambda de 10 GB	BUILD_LAMBDA_10GB	LINUX_LAMBDA_CONTAINER	10 GiB	10 GB

Ao usar outros tipos de ambiente, é recomendável usar uma imagem em cache para reduzir os tempos de compilação.

O espaço em disco listado para cada ambiente de compilação está disponível somente no diretório especificado pela variável de ambiente `CODEBUILD_SRC_DIR`.

Para escolher um tipo de computação:

- No CodeBuild console, no assistente Criar projeto de compilação ou na página Editar projeto de compilação, em Ambiente, expanda Configuração adicional e escolha uma das opções em Tipo

de computação. Para ter mais informações, consulte [Criar um projeto de compilação \(console\)](#) ou [Alterar configurações de um projeto de compilação \(console\)](#).

- Para o AWS CLI, execute o `update-project` comando `create-project` or, especificando o `computeType` valor do `environment` objeto. Para ter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#) ou [Alterar configurações de um projeto de compilação \(AWS CLI\)](#).
- Para o AWS SDKs, chame o equivalente da `UpdateProject` operação `CreateProject` ou para sua linguagem de programação de destino, especificando o equivalente ao `computeType` valor do `environment` objeto. Para obter mais informações, consulte o [AWS SDKs e referência de ferramentas](#).

Alguns tipos de ambiente e computação têm limitações de disponibilidade de região:

- O tipo de computação Linux GPU Small (`LINUX_GPU_CONTAINER`) só está disponível nas seguintes regiões:
 - Leste dos EUA (N. da Virgínia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Tóquio)
 - Canadá (Central)
 - Europa (Frankfurt)
 - Europa (Irlanda)
 - Europa (Londres)
- O tipo de computação Linux GPU Large (`LINUX_GPU_CONTAINER`) só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (N. da Virgínia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Seul)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Canadá (Central)
 - China (Pequim)
 - China (Ningxia)

- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- O modo de computação BUILD_GENERAL1_2XLARGE só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (Norte da Virgínia)
 - Oeste dos EUA (Norte da Califórnia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Hyderabad)
 - Ásia-Pacífico (Hong Kong)
 - Ásia-Pacífico (Jacarta)
 - Ásia-Pacífico (Melbourne)
 - Ásia-Pacífico (Mumbai)
 - Ásia-Pacífico (Seul)
 - Ásia-Pacífico (Singapura)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Canadá (Central)
 - China (Pequim)
 - China (Ningxia)
 - Europa (Frankfurt)
 - Europa (Irlanda)
 - Europa (Londres)
 - Europa (Paris)
 - Europa (Espanha)
 - Europa (Estocolmo)
 - Europa (Zurique)
 - Israel (Tel Aviv)
 - **Oriente Médio (Bahrein)**
- Oriente Médio (Emirados Árabes Unidos)

- América do Sul (São Paulo)
- O modo de ambiente ARM_CONTAINER só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (Norte da Virgínia)
 - Oeste dos EUA (Norte da Califórnia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Hong Kong)
 - Ásia-Pacífico (Jacarta)
 - Ásia-Pacífico (Hyderabad)
 - Ásia-Pacífico (Mumbai)
 - Ásia-Pacífico (Osaka)
 - Ásia-Pacífico (Seul)
 - Ásia-Pacífico (Singapura)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Canadá (Central)
 - China (Pequim)
 - China (Ningxia)
 - Europa (Frankfurt)
 - Europa (Irlanda)
 - Europa (Londres)
 - Europa (Milão)
 - Europe (Paris)
 - Europa (Espanha)
 - Europa (Estocolmo)
 - Israel (Tel Aviv)
 - Oriente Médio (Bahrein)
 - Oriente Médio (Emirados Árabes Unidos)
- América do Sul (São Paulo)

- O modo de ambiente `WINDOWS_SERVER_2022_CONTAINER` só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (N. da Virgínia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Europa (Frankfurt)
 - Europa (Irlanda)
 - América do Sul (São Paulo)
- O tipo de ambiente `LINUX_EC2` (`BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (Norte da Virgínia)
 - Oeste dos EUA (Norte da Califórnia)
 - Oeste dos EUA (Oregon)
 - África (Cidade do Cabo)
 - Ásia-Pacífico (Hong Kong)
 - Ásia-Pacífico (Jacarta)
 - Ásia-Pacífico (Melbourne)
 - Europa (Zurique)
 - Ásia-Pacífico (Hyderabad)
 - Ásia-Pacífico (Mumbai)
 - Ásia-Pacífico (Osaka)
 - Ásia-Pacífico (Seul)
 - Ásia-Pacífico (Singapura)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Canadá (Central)
 - China (Pequim)

- China (Ningxia)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europe (Paris)
- Europa (Espanha)
- Europa (Estocolmo)
- Israel (Tel Aviv)
- Oriente Médio (Bahrein)
- Oriente Médio (Emirados Árabes Unidos)
- América do Sul (São Paulo)
- AWS GovCloud (Oeste dos EUA)
- AWS GovCloud (Leste dos EUA)
- O tipo de ambiente ARM_EC2 (BUILD_GENERAL1_SMALL, BUILD_GENERAL1_MEDIUM, BUILD_GENERAL1_LARGE) só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (Norte da Virgínia)
 - Oeste dos EUA (Norte da Califórnia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Hong Kong)
 - Ásia-Pacífico (Jacarta)
 - Europa (Zurique)
 - Ásia-Pacífico (Hyderabad)
 - Ásia-Pacífico (Mumbai)
 - Ásia-Pacífico (Osaka)
 - Ásia-Pacífico (Seul)
 - Ásia-Pacífico (Singapura)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)

- Canadá (Central)
- China (Pequim)
- China (Ningxia)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europe (Paris)
- Europa (Espanha)
- Europa (Estocolmo)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (Oeste dos EUA)
- AWS GovCloud (Leste dos EUA)
- O tipo de ambiente `WINDOWS_EC2` (`BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) só está disponível nas seguintes regiões:
 - Leste dos EUA (Ohio)
 - Leste dos EUA (N. da Virgínia)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Sydney)
 - Ásia-Pacífico (Tóquio)
 - Europa (Frankfurt)
 - Europa (Irlanda)
 - América do Sul (São Paulo)
- O modo de computação AWS Lambda (`ARM_LAMBDA_CONTAINER` e `LINUX_LAMBDA_CONTAINER`) só está disponível nas seguintes regiões:
 - Leste dos EUA (Norte da Virgínia)
 - Leste dos EUA (Ohio)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Mumbai)

- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Europa (Frankfurt)
- Europa (Irlanda)
- América do Sul (São Paulo)
- O modo de computação MAC_ARM só está disponível nas seguintes regiões:
 - Leste dos EUA (Norte da Virgínia)
 - Leste dos EUA (Ohio)
 - Oeste dos EUA (Oregon)
 - Ásia-Pacífico (Sydney)
 - Europa (Frankfurt)

Para o tipo de computação BUILD_GENERAL1_2XLARGE, há suporte para imagens de Docker de até 100 GB descompactadas.

 Note

Para imagens de ambiente de compilação personalizadas, CodeBuild suporta imagens do Docker de até 50 GB descompactadas no Linux e no Windows, independentemente do tipo de computação. Para verificar o tamanho da imagem da compilação, use o Docker para executar o comando `docker images` *REPOSITORY:TAG*.

É possível usar o Amazon EFS para acessar mais espaço no contêiner de compilação. Para obter mais informações, consulte [Exemplo do Amazon Elastic File System para AWS CodeBuild](#). Se você deseja manipular espaço em disco do contêiner durante uma compilação, a compilação deverá ser executada no modo privilegiado.

 Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de](#)

[execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

Shells e comandos em ambientes de compilação

Você fornece um conjunto de comandos AWS CodeBuild para execução em um ambiente de compilação durante o ciclo de vida de uma compilação (por exemplo, instalando dependências de compilação e testando e compilando seu código-fonte). Há várias maneiras de especificar esses comandos:

- Crie um arquivo de especificação da compilação e inclua-o no código-fonte. Nesse arquivo, especifique os comandos que deseja executar em cada fase de ciclo de vida do build. Para obter mais informações, consulte [Referência de especificação de construção para CodeBuild](#).
- Use o CodeBuild console para criar um projeto de compilação. Em Insert build commands (Inserir comandos de compilação), para Build commands (Comandos de compilação), insira os comandos que deseja executar na fase build. Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#).
- Use o CodeBuild console para alterar as configurações de um projeto de compilação. Em Insert build commands (Inserir comandos de compilação), para Build commands (Comandos de compilação), insira os comandos que deseja executar na fase build. Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#).
- Use o AWS CLI ou AWS SDKs para criar um projeto de compilação ou alterar as configurações de um projeto de compilação. Faça referência ao código-fonte que contenha um arquivo de buildspec com os comandos ou especifique uma única string que inclua o conteúdo de um arquivo de buildspec equivalente. Para ter mais informações, consulte [Criar um projeto de compilação](#) ou [Alterar as configurações do projeto de compilação](#).
- Use o AWS CLI or AWS SDKs para iniciar uma compilação, especificando um arquivo buildspec ou uma única string que inclua o conteúdo de um arquivo buildspec equivalente. Para obter mais informações, consulte a descrição para o valor buildspecOverride em [Executar compilações manualmente](#).

É possível especificar qualquer comando da linguagem de comandos Shell (sh). Na versão 0.1 do buildspec, CodeBuild executa cada comando do Shell em uma instância separada no ambiente de compilação. Isso significa que cada comando é executado isoladamente em relação aos outros comandos. Portanto, por padrão, não é possível executar um comando que dependa do estado

de algum comando anterior (por exemplo, mudança de diretórios ou configuração de variáveis de ambiente). Para resolver essa limitação, recomendamos usar a versão 0.2, que resolve o problema. Caso você precise usar a versão 0.1, recomendamos as seguintes abordagens:

- Inclua um script de shell em seu código-fonte que contenha os comandos que você deseja executar em uma única instância do shell padrão. Por exemplo, você poderia incluir um arquivo denominado `my-script.sh` em seu código-fonte que contenha comandos como `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd; .`. Depois, no arquivo de buildspec, especifique o comando `./my-script.sh`.
- No arquivo de buildspec, ou na configuração Build commands (Comandos de compilação) apenas da fase build, especifique um único comando que inclua todos os comandos que você deseja executar em uma única instância do shell padrão (por exemplo, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

Se CodeBuild encontrar um erro, pode ser mais difícil solucionar o erro em comparação com a execução de um único comando em sua própria instância do shell padrão.

Os comandos que são executados em uma imagem do Windows Server Core usam o PowerShell shell.

Variáveis de ambiente em ambientes de compilação

AWS CodeBuild fornece várias variáveis de ambiente que você pode usar em seus comandos de construção:

AWS_DEFAULT_REGIÃO

A AWS região em que a compilação está sendo executada (por exemplo, `us-east-1`). Essa variável de ambiente é usada principalmente pelo AWS CLI.

AWS_REGION

A AWS região em que a compilação está sendo executada (por exemplo, `us-east-1`). Essa variável de ambiente é usada principalmente pelo AWS SDKs.

CODEBUILD_BATCH_BUILD_IDENTIFIER

O identificador da compilação em uma compilação em lote. É determinado na especificação de compilação em lote. Para obter mais informações, consulte [the section called “Referência de buildspec em lote”](#).

CODEBUILD_BUILD_ARN

O nome do recurso da Amazon (ARN) da compilação (por exemplo, `arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

O CodeBuild ID da compilação (por exemplo, `codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

O identificador da imagem de CodeBuild construção (por exemplo, `aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

O número de compilação atual do projeto.

CODEBUILD_BUILD_SUCCEEDING

Se a compilação atual foi bem-sucedida. Defina como 0 caso a compilação falhe, ou 1 caso seja bem-sucedida.

CODEBUILD_INITIATOR

A entidade que iniciou a compilação. Se CodePipeline a compilação foi iniciada, esse é o nome do pipeline (por exemplo, `codepipeline/my-demo-pipeline`). Se um usuário iniciou a compilação, é o nome do usuário (por exemplo, `MyUserName`). Se o plug-in Jenkins CodeBuild iniciou a compilação, essa é a `stringCodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

O identificador da AWS KMS chave que CodeBuild está sendo usada para criptografar o artefato de saída da compilação (por exemplo, `arn:aws:kms:region-ID:account-ID:key/key-ID ou alias/key-alias`).

CODEBUILD_PROJECT_ARN

O Amazon Resource Name (ARN) do projeto (por exemplo, `arn:aws:codebuild:region-ID:account-ID:project/project-name`).

CODEBUILD_PUBLIC_BUILD_URL

O URL dos resultados dessa compilação no site de compilações públicas. Essa variável só será definida se o projeto de compilação tiver compilações públicas habilitadas. Para obter mais informações, consulte [Obtenha um projeto de construção público URLs](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

O identificador de versão de um código-fonte de uma compilação. O conteúdo depende do repositório do código-fonte:

CodeCommit, GitHub, GitHub Enterprise Server e Bitbucket

Essa variável contém o ID de confirmação.

CodePipeline

Essa variável contém a revisão da fonte fornecida por CodePipeline.

Se não CodePipeline for possível resolver a revisão da fonte, como quando a origem é um bucket do Amazon S3 que não tem o versionamento ativado, essa variável de ambiente não está definida.

Amazon S3

Essa variável não está definida.

Quando aplicável, a variável CODEBUILD_RESOLVED_SOURCE_VERSION só está disponível após a fase DOWNLOAD_SOURCE.

CODEBUILD_SOURCE_REPO_URL

O URL do artefato de entrada ou repositório de código-fonte. Para o Amazon S3, é `s3://` seguido do nome do bucket e o caminho para o artefato de entrada. Para CodeCommit e GitHub, esse é o URL do clone do repositório. Se uma compilação for originada de CodePipeline, essa variável de ambiente pode estar vazia.

No caso de origens secundárias, a variável de ambiente do URL do repositório de origem secundária é CODEBUILD_SOURCE_REPO_URL_*<sourceIdentifier>*, em que *<sourceIdentifier>* é o identificador de origem criado.

CODEBUILD_SOURCE_VERSION

O formato do valor depende do repositório de origem.

- Para o Amazon S3, é o ID de versão associado ao artefato de entrada.

- Pois CodeCommit, é o ID do commit ou nome da ramificação associado à versão do código-fonte a ser criado.
- Para GitHub o GitHub Enterprise Server e o Bitbucket, é o ID do commit, o nome da ramificação ou o nome da tag associado à versão do código-fonte a ser criado.

**Note**

Para uma compilação GitHub ou GitHub Enterprise Server que é acionada por um evento de pull request de webhook, é `pull-request-number`.

No caso de origens secundárias, a variável de ambiente da versão de origem secundária é `CODEBUILD_SOURCE_VERSION_<sourceIdentifier>`, em que `<sourceIdentifier>` é o identificador de origem criado. Para obter mais informações, consulte [Várias fontes de entrada e exemplos de artefatos de saída](#).

CODEBUILD_SRC_DIR

O caminho do diretório CodeBuild usado para a compilação (por exemplo, `/tmp/src123456789/src`).

No caso de origens secundárias, a variável de ambiente do caminho de diretório de origem secundária é `CODEBUILD_SRC_DIR_<sourceIdentifier>`, em que `<sourceIdentifier>` é o identificador de origem criado. Para obter mais informações, consulte [Várias fontes de entrada e exemplos de artefatos de saída](#).

CODEBUILD_START_TIME

A hora de início da compilação especificada como um carimbo de data e hora do Unix em milissegundos.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

O ID da conta do usuário que acionou o evento webhook.

CODEBUILD_WEBHOOK_BASE_REF

O nome da referência de base do evento webhook que aciona a compilação atual. Para uma solicitação pull, isso é a referência da ramificação.

CODEBUILD_WEBHOOK_EVENT

O evento webhook que aciona a compilação atual.

CODEBUILD_WEBHOOK_MERGE_COMMIT

O identificador de confirmação de mesclagem usado para a compilação. Essa variável é definida quando uma solicitação pull do Bitbucket é mesclada com a estratégia squash e a ramificação da solicitação pull é fechada. Nesse caso, a confirmação de solicitação pull original não existe mais, portanto, a variável de ambiente contém o identificador da confirmação de mesclagem comprimida.

CODEBUILD_WEBHOOK_PREV_COMMIT

O ID da confirmação mais recente antes do evento de push de webhook que aciona a compilação atual.

CODEBUILD_WEBHOOK_HEAD_REF

O nome da referência head do evento webhook que aciona a compilação atual. Ele pode ser uma ramificação de referência ou uma tag de referência.

CODEBUILD_WEBHOOK_TRIGGER

Mostra o evento webhook que acionou a compilação. Essa variável está disponível apenas para compilações acionadas por um webhook. O valor é analisado a partir da carga enviada para CodeBuild by GitHub, GitHub Enterprise Server ou Bitbucket. O formato do valor depende de qual tipo de evento acionou a compilação.

- Para compilações acionadas por uma solicitação pull, é `pr/pull-request-number`.
- Para compilações acionadas por meio da criação de uma nova ramificação ou envio de uma confirmação para uma ramificação, é `branch/branch-name`.
- Para compilações acionadas pelo envio de uma tag a um repositório, é `tag/tag-name`.

HOME

Essa variável de ambiente é sempre configurada como `/root`.

AWS CodeBuild também oferece suporte a um conjunto de variáveis de ambiente para compilações de executores auto-hospedados. Para saber mais sobre o executor CodeBuild auto-hospedado, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#)

CODEBUILD_RUNNER_OWNER

O proprietário do repositório que aciona a compilação do executor auto-hospedado.

CODEBUILD_RUNNER_REPO

O nome do repositório que aciona a compilação do executor auto-hospedado.

CODEBUILD_RUNNER_REPO_DOMAIN

O domínio do repositório que aciona a compilação do executor auto-hospedado. Somente compilações GitHub Enterprise especificadas.

CODEBUILD_WEBHOOK_LABEL

O rótulo usado para configurar as substituições de compilação e o executor auto-hospedado durante a compilação.

CODEBUILD_WEBHOOK_RUN_ID

O ID de execução do fluxo de trabalho associado à compilação.

CODEBUILD_WEBHOOK_JOB_ID

O ID do trabalho associado à compilação.

CODEBUILD_WEBHOOK_WORKFLOW_NAME

O nome do fluxo de trabalho associado à compilação, se existir na carga útil da solicitação do webhook.

CODEBUILD_RUNNER_WITH_BUILDSPEC

Se uma substituição de buildspec estiver configurada nos rótulos de solicitação do executor auto-hospedado, isso será definido como `true`.

Você também pode fornecer ambientes de build com suas próprias variáveis de ambiente. Para obter mais informações, consulte os tópicos a seguir.

- [Use CodeBuild com CodePipeline](#)
- [Criar um projeto de compilação](#)
- [Alterar as configurações do projeto de compilação](#)
- [Executar compilações manualmente](#)
- [Referência de buildspec](#)

Para exibir todas as variáveis de ambiente disponíveis em um ambiente de compilação, você pode executar o comando `printenv` (em ambientes de compilação baseados no Linux) ou o comando

"Get-ChildItem Env:" (em ambientes de compilação baseados no Windows) durante uma compilação. Com exceção das listadas anteriormente, as variáveis de ambiente que começam com CODEBUILD_ são para uso CodeBuild interno. Elas não devem ser usadas em seus comandos de build.

Important

Nós desencorajamos fortemente o uso de variáveis de ambiente para armazenar valores confidenciais, especialmente a chave IDs de AWS acesso. As variáveis de ambiente podem ser exibidas em texto simples usando ferramentas como o CodeBuild console e AWS CLI o. Recomendamos que você armazene valores confidenciais no Amazon EC2 Systems Manager Parameter Store e, em seguida, recupere-os do seu buildspec. Para armazenar valores confidenciais, consulte [Armazenamento de parâmetros e instruções passo a passo do Systems Manager: Crie e teste um parâmetro de string \(console\) no Guia](#) do usuário do Amazon EC2 Systems Manager. Para recuperá-los, consulte o mapeamento parameter-store em [Sintaxe de buildspec](#).

CODEBUILD_BUILD_URL

O URL dos resultados da compilação para essa compilação.

Tarefas em segundo plano em ambientes de compilação

Você pode executar tarefas em segundo plano em ambientes de compilação. Para isso, na especificação de compilação, use o comando `nohup` para executar um comando como uma tarefa em segundo plano, mesmo se o processo de compilação sair do shell. Use o comando `disown` para interromper forçadamente uma tarefa em execução em segundo plano.

Exemplos:

- Inicie um processo em segundo plano e aguarde a conclusão dele mais tarde:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Inicie um processo em segundo plano e não aguarde a conclusão dele:

```
|  
nohup sleep 30 & disown $!
```

- Inicie um processo em segundo plano e o encerre depois:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
kill $(cat pidfile)
```

Projetos de compilação

Um projeto de compilação inclui informações sobre como executar uma compilação, incluindo onde obter o código-fonte, qual ambiente de compilação usar, quais comandos de compilação executar e onde armazenar a saída da compilação.

Você pode executar estas tarefas para trabalhar com projetos de build:

Tópicos

- [Crie um projeto de construção em AWS CodeBuild](#)
- [Criar uma regra de notificação](#)
- [Alterar as configurações do projeto de construção em AWS CodeBuild](#)
- [Vários tokens de acesso em CodeBuild](#)
- [Excluir projetos de compilação em AWS CodeBuild](#)
- [Obtenha um projeto de construção público URLs](#)
- [Compartilhar projetos de compilação](#)
- [Projetos de compilação de tags](#)
- [Use corredores com AWS CodeBuild](#)
- [Use webhooks com AWS CodeBuild](#)
- [Veja os detalhes de um projeto de construção em AWS CodeBuild](#)
- [Veja os nomes dos projetos de construção em AWS CodeBuild](#)

Crie um projeto de construção em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para criar um projeto de construção.

Tópicos

- [Pré-requisitos](#)
- [Criar um projeto de compilação \(console\)](#)
- [Criar um projeto de compilação \(AWS CLI\)](#)
- [Criar um projeto de compilação \(AWS SDKs\)](#)
- [Criar um projeto de compilação \(AWS CloudFormation\)](#)

Pré-requisitos

Antes de criar um projeto de compilação, responda às perguntas em [Planejar uma compilação](#).

Criar um projeto de compilação (console)

Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.

Selecione Create build project (Criar projeto de compilação).

Preencha as seções a seguir. Depois de concluir, escolha Criar projeto de compilação na parte inferior da página.

Seções:

- [Configuração de projetos](#)
- [Origem](#)
- [Environment](#)
- [Buildspec](#)
- [Configuração em lote](#)
- [Artefatos](#)
- [Logs](#)

Configuração de projetos

Nome do projeto

Insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta.

Descrição

Insira uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.

Emblema da compilação

Selecione Habilitar emblema da compilação para que o status de compilação do projeto fique visível e possa ser incorporado. Para obter mais informações, consulte [Exemplo de emblemas de compilação](#).

Note

O emblema de compilação não se aplicará se o provedor de origem for Amazon S3.

Habilitar limite de compilações simultâneas

(Opcional) Se você quiser limitar o número de compilações simultâneas para esse projeto, execute as seguintes etapas:

1. Selecione Restringir número de compilações simultâneas que este projeto pode iniciar.
2. Em Limite de compilações simultâneas, insira o número máximo de compilações simultâneas permitidas para esse projeto. Esse limite não pode ser maior que o limite de compilações simultâneas definido para a conta. Se você tentar inserir um número maior que o limite da conta, uma mensagem de erro será exibida.

Novas compilações só são iniciadas se o número atual de compilações for menor ou igual a esse limite. Se a contagem de compilações atual atender a esse limite, novas compilações serão limitadas e não serão executadas.

Informações adicionais

(Opcional) Em Tags, insira o nome e o valor de todas as tags que você deseja que AWS os serviços de suporte usem. Use Adicionar linha para adicionar uma tag. É possível adicionar até 50 tags.

Origem

Provedor de origem

Escolha o tipo de provedor do código-fonte. Use as seguintes listas para fazer seleções adequadas ao provedor de origem:

 Note

CodeBuild não é compatível com o Bitbucket Server.

Amazon S3

Bucket

Escolha o nome do bucket de entrada que contém o código-fonte.

Chave de objeto do S3 ou pasta do S3

Insira o nome do arquivo ZIP ou o caminho na pasta que contém o código-fonte. Insira uma barra (/) para fazer download de tudo no bucket do S3.

Versão de origem

Insira o ID da versão do objeto que representa a compilação do arquivo de entrada. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

CodeCommit

Repositório

Selecione o repositório que deseja usar.

Tipo de referência

Selecione Ramificação, Tag do Git ou ID de confirmação para especificar a versão do código-fonte. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

 Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Bitbucket

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha CodeConnections a OAuth senha do aplicativo ou o token de acesso pessoal para se conectar CodeBuild.

Conexão

Selecione uma conexão do Bitbucket ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório na minha conta Bitbucket ou repositório público e insira o URL do repositório.

Versão de origem

Insira uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para ter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#)

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o parâmetro name no status de confirmação do Bitbucket. Para obter mais informações, consulte [compilação](#) na documentação da API do Bitbucket.

Em URL de destino, insira o valor a ser usado para o parâmetro url no status de confirmação do Bitbucket. Para obter mais informações, consulte [compilação](#) na documentação da API do Bitbucket.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [Filtrar eventos de webhook do Bitbucket](#).

GitHub

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha GitHub Aplicativo ou Token de acesso pessoal para se conectar CodeBuild. OAuth

Conexão

Selecione uma GitHub conexão ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório em minha GitHub conta, repositório público ou webhook com GitHub escopo definido e insira a URL do repositório.

Versão de origem

Insira uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para ter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#)

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o context parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

Em URL de destino, insira o valor a ser usado para o target_url parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [GitHub eventos de webhook](#).

GitHub Enterprise Server

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha CodeConnectionso token de acesso pessoal ao qual se conectar CodeBuild.

Conexão

Selecione uma conexão GitHub Enterprise ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório na minha conta GitHub corporativa ou webhook com escopo GitHub corporativo e insira a URL do repositório.

Versão de origem

Insira uma solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o `context` parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

Em URL de destino, insira o valor a ser usado para o `target_url` parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

SSL inseguro

Selecione Habilitar SSL inseguro para ignorar os avisos de SSL ao se conectar ao seu GitHub repositório de projetos corporativos.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [GitHub eventos de webhook](#).

GitLab

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

CodeConnections é usado para se GitLab conectar CodeBuild a.

Conexão

Selecione uma GitLab conexão pela qual se conectar CodeConnections.

Repositório

Selecione o repositório que deseja usar.

Versão de origem

Insira um ID de solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

GitLab Self Managed

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

CodeConnectionsé usado para conectar o GitLab Self Managed CodeBuild a.

Conexão

Selecione uma conexão GitLab autogerenciada para se conectar CodeConnections.

Repositório

Selecione o repositório que deseja usar.

Versão de origem

Insira um ID de solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Environment

Modelo de provisionamento

Execute um destes procedimentos:

- Para usar frotas sob demanda gerenciadas por AWS CodeBuild, escolha Sob demanda. Com frotas sob demanda, CodeBuild fornece computação para suas construções. As máquinas são destruídas quando a compilação termina. As frotas sob demanda são totalmente gerenciadas e incluem recursos de escalabilidade automática para lidar com picos de demanda.
- Para usar frotas de capacidade reservada gerenciadas por AWS CodeBuild, escolha Capacidade reservada e selecione um nome de frota. Com frotas de capacidade reservada, você configura um conjunto de instâncias dedicadas para seu ambiente de compilação. Essas máquinas permanecem ociosas, prontas para processar compilações ou testes imediatamente e reduzem a duração da compilação. Com frotas de capacidade reservada, suas máquinas estão sempre funcionando e continuarão a incorrer em custos enquanto forem provisionadas.

Para ter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#).

Imagem do ambiente

Execute um destes procedimentos:

- Para usar uma imagem do Docker gerenciada por AWS CodeBuild, escolha Imagem gerenciada e, em seguida, faça seleções em Sistema operacional, Tempo de execução (s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.
- Para usar outra imagem do Docker, selecione Custom image (Imagem personalizada). Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS
- Para usar uma imagem privada do Docker, selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

Note

CodeBuild substitui as imagens personalizadas do ENTRYPOINT Docker.

Computação

Execute um destes procedimentos:

- Para usar a EC2 computação, escolha EC2. EC2 a computação oferece flexibilidade otimizada durante a execução de ações.
- Para usar a computação Lambda, escolha Lambda. A computação Lambda oferece velocidades de inicialização otimizadas para as compilações. O Lambda oferece suporte a compilações mais rápidas devido à menor latência de inicialização. O Lambda também é escalado automaticamente, portanto as compilações não aguardam na fila para serem executadas. Para ter mais informações, consulte [Execute compilações em computação AWS Lambda](#).

Perfil de serviço

Execute um destes procedimentos:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

Note

Ao usar o console para criar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

Configuração adicional

Limite de repetição automática

Especifique o número de repetições automáticas adicionais após a falha de uma compilação. Por exemplo, se o limite de repetição automática estiver definido como 2, CodeBuild chamará a RetryBuild API para repetir automaticamente sua compilação por até 2 vezes adicionais.

Timeout (Tempo limite)

Especifique um valor, entre 5 minutos e 36 horas, após o qual CodeBuild interrompe a compilação se ela não for concluída. Se hours e minutes forem deixados em branco, o valor padrão de 60 minutos será usado.

Privilegiado

(Opcional) Selecione Ativar este sinalizador se quiser criar imagens do Docker ou desejar que suas compilações obtenham privilégios elevados somente se planeja usar esse projeto de compilação para criar imagens do Docker. Do contrário, todas as compilações associadas que tentarem interagir com o daemon do Docker falharão. Você também deve iniciar o daemon do Docker, de maneira que as compilações possam interagir com ele. Uma maneira de fazer isso é inicializar o daemon do Docker na fase `install` de especificação da compilação executando os comandos de compilação a seguir. Não execute esses comandos se você escolher uma imagem do ambiente de compilação fornecida CodeBuild com o suporte do Docker.

Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Se você quiser CodeBuild trabalhar com sua VPC:

- Para VPC, escolha a ID da VPC que usa. CodeBuild
- Para sub-redes VPC, escolha as sub-redes que incluem os recursos que usa. CodeBuild
- Para grupos de segurança da VPC, escolha os grupos de segurança CodeBuild usados para permitir o acesso aos recursos no. VPCs

Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

Computação

Selecione uma das opções disponíveis.

Credencial de registro

Especifique uma credencial de registro quando o projeto estiver configurado com uma imagem de registro não privada.

Note

Essa credencial só será utilizada se as imagens forem substituídas pelas de registros privados.

Variáveis de ambiente

Insira o nome e o valor e, depois, selecione o tipo de cada variável de ambiente a ser usada pelas compilações.

Note

CodeBuild define automaticamente a variável de ambiente para sua AWS região. Defina as seguintes variáveis de ambiente se elas não tiverem sido adicionadas ao `buildspec.yml`:

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

O console e AWS CLI os usuários podem ver as variáveis de ambiente. Caso você não tenha problema em relação à visibilidade da variável de ambiente, defina os campos Name e Value e Type como Plaintext.

Recomendamos que você armazene uma variável de ambiente com um valor confidencial, como um ID de chave de AWS acesso, uma chave de acesso AWS secreta ou uma senha

como parâmetro no Amazon EC2 Systems Manager Parameter Store ou AWS Secrets Manager.

Se você usa o Amazon EC2 Systems Manager Parameter Store, em Tipo, escolha Parâmetro. Em Nome, insira um identificador CodeBuild para referência. Em Value, insira o nome do parâmetro conforme armazenado no Amazon EC2 Systems Manager Parameter Store. Usando um parâmetro chamado `/CodeBuild/dockerLoginPassword` como exemplo, em Type (Tipo), escolha Parameter (Parâmetro). Em Nome, digite `LOGIN_PASSWORD`. Em Valor, insira `/CodeBuild/dockerLoginPassword`.

Important

Se você usa o Amazon EC2 Systems Manager Parameter Store, recomendamos que você armazene parâmetros com nomes de parâmetros que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Você pode usar o CodeBuild console para criar um parâmetro no Amazon EC2 Systems Manager. Selecione Create parameter (Criar parâmetro) e siga as instruções na caixa de diálogo. (Nessa caixa de diálogo, para a chave KMS, você pode especificar o ARN de AWS KMS uma chave na sua conta. O Amazon EC2 Systems Manager usa essa chave para criptografar o valor do parâmetro durante o armazenamento e descriptografá-lo durante a recuperação.) Se você usar o CodeBuild console para criar um parâmetro, o console iniciará o nome do parâmetro com a `/CodeBuild/` forma como ele está sendo armazenado. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store, a função de serviço do projeto de construção deve permitir a `ssm:GetParameters` ação. Se você escolheu Nova função de serviço anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store com nomes de parâmetros que não começam com `/CodeBuild/`, e você escolheu Nova função de serviço, você deve atualizar essa função de serviço para permitir acesso aos nomes de parâmetros que não

começam com `/CodeBuild/`. Isso porque essa função de serviço permite acesso apenas a nomes de parâmetro que comecem com `/CodeBuild/`.

Se você escolher Nova função de serviço, a função de serviço incluirá permissão para descriptografar todos os parâmetros sob o `/CodeBuild/` namespace no Amazon Systems EC2 Manager Parameter Store.

As variáveis de ambiente definidas por você substituem variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você definir uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já contiver uma variável de ambiente chamada `PATH` com um valor de `/usr/local/sbin:/usr/local/bin` e você definir uma variável de ambiente chamada `PATH` com um valor de `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`. Não defina nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome é definida em vários locais, o valor será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade.
- O valor na definição de projeto de build tem a precedência seguinte.
- O valor na declaração de buildspec tem a menor prioridade.

Se você usar o Secrets Manager, em Tipo, escolha Secrets Manager. Em Nome, insira um identificador CodeBuild para referência. Em Value (Valor), insira um `reference-key` usando o padrão `secret-id:json-key:version-stage:version-id`. Para ter mais informações, consulte [Secrets Manager reference-key in the buildspec file](#).

Important

Se você usa o Secrets Manager, recomendamos armazenar segredos com nomes que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

Se o projeto de compilação se referir a segredos armazenados no Secrets Manager, o perfil de serviço do projeto de compilação deverá permitir a ação

`secretsmanager:GetSecretValue`. Se você escolheu Nova função de serviço anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se o projeto de compilação fizer referência a segredos armazenados no Secrets Manager com nomes que não comecem com `/CodeBuild/` e você selecionar Novo perfil de serviço, será necessário atualizar esse perfil de serviço para conceder acesso a nomes de segredo que não comecem com `/CodeBuild/`. O motivo é que o perfil de serviço permite acesso apenas a nomes de segredo que comecem com `/CodeBuild/`.

Se você selecionar Novo perfil de serviço, o perfil de serviço incluirá permissão para descriptografar todos os segredos no namespace `/CodeBuild/` no Secrets Manager.

Buildspec

Especificações de compilação

Execute um destes procedimentos:

- Se o seu código-fonte inclui um arquivo buildspec, escolha Use a buildspec file (Usar um arquivo buildspec). Por padrão, o CodeBuild procura um arquivo chamado `buildspec.yml` no diretório raiz do código-fonte. Se o arquivo buildspec usar um nome ou um local diferente, insira o caminho a partir da raiz de origem em Nome do buildspec (por exemplo, `buildspec-two.yml` ou `configuration/buildspec.yml`). Se o arquivo buildspec estiver em um bucket do S3, ele deverá estar na mesma região da AWS do projeto de compilação. Especifique o arquivo buildspec usando o ARN (por exemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Se o código-fonte não incluir um arquivo de especificação de compilação ou se você quiser executar comandos de compilação diferentes dos especificados para a fase build no arquivo `buildspec.yml`, no diretório raiz do código-fonte, escolha Insert build commands. Para Build commands (Comandos de compilação), insira os comandos que você quer executar na fase build. Para vários comandos, separe-os com `&&` (por exemplo, `mvn test && mvn package`). Para executar comandos em outras fases, ou se você tiver uma lista longa de comandos para a fase build, adicione um arquivo `buildspec.yml` ao diretório raiz do código-fonte, adicione os comandos ao arquivo e escolha Usar o buildspec.yml no diretório raiz do código-fonte.

Para obter mais informações, consulte [Referência de buildspec](#).

Configuração em lote

É possível executar um grupo de compilações como uma única operação. Para obter mais informações, consulte [Executar compilações em lotes](#).

Definir a configuração em lote

Selecione para permitir compilações em lote neste projeto.

Perfil de serviço em lote

Fornece o perfil de serviço para compilações em lote.

Escolha uma das seguintes opções:

- Se você não tiver um perfil de serviço em lote, selecione Novo perfil de serviço. Em Perfil de serviço, insira um nome para o novo perfil.
- Se você tiver um perfil de serviço em lote, selecione Perfil de serviço existente. Em Perfil de serviço, selecione o perfil de serviço.

As compilações em lote introduzem um novo perfil de segurança na configuração em lote. Essa nova função é necessária, pois CodeBuild você deve poder chamar as `RetryBuild` ações `StartBuild``StopBuild`, e em seu nome para executar compilações como parte de um lote. Os clientes devem usar um novo perfil, e não o mesmo perfil que usam na compilação, por dois motivos:

- Fornecer ao perfil de compilação as permissões `StartBuild`, `StopBuild` e `RetryBuild` que permitem a uma única compilação iniciar mais compilações por meio do `buildspec`.
- CodeBuild compilações em lote fornecem restrições que restringem o número de compilações e tipos de computação que podem ser usados para as compilações no lote. Se o perfil de compilação tiver essas permissões, será possível que as próprias compilações ignorem essas restrições.

Tipos de computação permitidos para lote

Selecione os tipos de computação permitidos para o lote. Selecione todas as opções aplicáveis.

Frotas permitidas por lote

Selecione as frotas permitidas para o lote. Selecione todas as opções aplicáveis.

Máximo de compilações permitidas em lote

Insira o número máximo de compilações permitidas no lote. Se um lote exceder esse limite, ele falhará.

Tempo limite do lote

Insira o tempo máximo para a conclusão da compilação em lote.

Combinar artefatos

Selecione Combinar todos os artefatos do lote em um único local para que todos os artefatos do lote sejam combinados em um único local.

Modo de relatório em lote

Selecione o modo de relatório de status de compilação desejado para compilações em lote.

Note

Esse campo só está disponível quando a fonte do projeto é Bitbucket ou GitHub Enterprise GitHub, e Relatar os status de compilação ao provedor de origem quando suas compilações começam e terminam é selecionado em Fonte.

Compilações agregadas

Selecione para que os status de todas as compilações do lote sejam combinados em um único relatório de status.

Compilações individuais

Selecione para que os status de todas as compilações no lote sejam relatados separadamente.

Artefatos

Tipo

Execute um destes procedimentos:

- Se você não quiser criar artefatos de saída de build, escolha No artifacts. É recomendável fazer isso caso esteja executando apenas testes de compilação ou queira enviar uma imagem do Docker a um repositório do Amazon ECR.

- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:
 - Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. (Se quiser um arquivo ZIP como saída e quiser que esse arquivo tenha uma extensão de arquivo, não se esqueça de incluir a extensão depois do nome do arquivo.)
 - Selecione Enable semântico versioning (Habilitar versionamento semântico) se quiser que um nome especificado no arquivo buildspec substitua qualquer nome especificado no console. O nome em um arquivo buildspec é calculado no tempo de compilação e usa a linguagem de comandos do Shell. Por exemplo, você pode anexar uma data e hora ao nome do artefato para que ele seja sempre exclusivo. Os nomes de artefato exclusivos impedem que os artefatos sejam substituídos. Para obter mais informações, consulte [Sintaxe de buildspec](#).
- Para Bucket name, selecione o nome do bucket de saída.
- Se você tiver escolhido Insert build commands (Inserir comandos de compilação) antes neste procedimento, em Output files (Arquivos de saída), insira os locais dos arquivos da compilação que deseja incluir no arquivo ZIP de saída da compilação ou na pasta. Para vários locais, separe-os com uma vírgula (por exemplo, appspec.yml, target/my-app.jar). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).
- Se você não quiser que os artefatos de compilação sejam criptografados, selecione Remove artifacts encryption (Remover a criptografia dos artefatos).

Para cada conjunto secundário de artefatos desejado:

1. Em Artifact identifier (Identificador de artefato), insira um valor com menos de 128 caracteres e que contenha apenas caracteres alfanuméricos e sublinhados.
2. Escolha Add artifact (Adicionar artefato).
3. Siga as etapas anteriores para configurar seus artefatos secundários.
4. Escolha Save artifact (Salvar artefato).

Configuração adicional

Chave de criptografia

(Opcional) Siga um destes procedimentos:

- Para usar a Chave gerenciada pela AWS para o Amazon S3 na conta para criptografar os artefatos de saída de compilação, deixe Chave de criptografia em branco. Esse é o padrão.
- Para usar uma chave gerenciada pelo cliente para criptografar os artefatos de saída da compilação, em Chave de criptografia, insira o ARN da chave do KMS. Use o formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo de cache

Em Cache type (Tipo de cache), selecione uma das seguintes opções:

- Se não quiser usar um cache, escolha No cache.
- Se você quiser um cache do Amazon S3, selecione Amazon S3 e faça o seguinte:
 - Em Bucket, escolha o nome do bucket do S3 onde o cache está armazenado.
 - (Opcional) Em Prefixo do caminho do cache, insira um prefixo de caminho do Amazon S3. O valor Cache path prefix (Prefixo do caminho do cache) é semelhante ao nome de um diretório. Ele permite que você armazene o cache em um bucket no mesmo diretório.

Important

Não acrescente uma barra (/) ao final do prefixo do caminho.

- Se você quiser usar um cache local, selecione Local e depois selecione um ou mais modos de cache local.

Note

O modo de cache de camada do Docker está disponível apenas para o Linux. Se optar por esse modo, o projeto deverá ser executado no modo privilegiado.

O uso do cache economiza um tempo de compilação considerável porque as partes reutilizáveis do ambiente de compilação são armazenadas no cache e usadas em diferentes compilações. Para obter informações sobre como especificar um cache no arquivo de especificação de compilação, consulte [Sintaxe de buildspec](#). Para obter mais informações sobre armazenamento em cache, consulte [Compilações em cache para melhorar o desempenho](#).

Logs

Selecione os logs que deseja criar. Você pode criar Amazon CloudWatch Logs, Amazon S3 logs ou ambos.

CloudWatch

Se você quiser CloudWatch registros do Amazon Logs:

CloudWatch logs

Selecione CloudWatch logs (Logs do &CW;).

Group name

Insira o nome do seu grupo de CloudWatch logs do Amazon Logs.

Nome do fluxo

Insira o nome do seu stream de CloudWatch log do Amazon Logs.

S3

Se você quiser que o Amazon S3 registre:

Logs do S3

Selecione S3 logs (Logs do S3).

Bucket

Escolha o nome do bucket do S3 para os logs.

Prefixo do caminho

Insira o prefixo para seus logs.

Desabilitar a criptografia de logs do S3

Selecione se não quiser que os logs do S3 sejam criptografados.

Criar um projeto de compilação (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with CodeBuild, consulte [Referência da linha de comando](#) o.

Para criar um projeto de CodeBuild construção usando o AWS CLI, você cria uma estrutura de [projeto](#) formatada em JSON, preenche a estrutura e chama o [create-project](#) comando para criar o projeto.

Criar o arquivo JSON

Crie um arquivo JSON esqueleto com o comando [create-project](#), usando a opção `--generate-cli-skeleton`:

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Isso cria um arquivo JSON com o caminho e o nome do arquivo especificados por *<json-file>*.

Preencher o arquivo JSON

Modifique os dados JSON da maneira a seguir e salve os resultados.

```
{
  "name": "<project-name>",
  "description": "<description>",
  "source": {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" | "GITLAB" |
    "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "buildStatusConfig": {
      "context": "<context>",
      "targetUrl": "<target-url>"
    },
    "gitSubmodulesConfig": {
      "fetchSubmodules": "<fetch-submodules>"
    },
    "auth": {
      "type": "<auth-type>",
      "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
  },
  "secondarySources": [
    {
```

```

    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
"GITLAB" | "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "auth": {
        "type": "<auth-type>",
        "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
}
],
"secondarySourceVersions": [
{
    "sourceIdentifier": "<secondary-source-identifier>",
    "sourceVersion": "<secondary-source-version>"
}
],
"sourceVersion": "<source-version>",
"artifacts": {
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<artifacts-location>",
    "path": "<artifacts-path>",
    "namespaceType": "<artifacts-namespacetype>",
    "name": "<artifacts-name>",
    "overrideArtifactName": "<override-artifact-name>",
    "packaging": "<artifacts-packaging>"
},
"secondaryArtifacts": [
{
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<secondary-artifact-location>",
    "path": "<secondary-artifact-path>",
    "namespaceType": "<secondary-artifact-namespaceType>",
    "name": "<secondary-artifact-name>",
    "packaging": "<secondary-artifact-packaging>",
    "artifactIdentifier": "<secondary-artifact-identifier>"
}
],
"cache": {
    "type": "<cache-type>",
    "location": "<cache-location>",

```

```

    "mode": [
        "<cache-mode>"
    ],
    "environment": {
        "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
"WINDOWS_SERVER_2019_CONTAINER" | "WINDOWS_SERVER_2022_CONTAINER",
        "image": "<image>",
        "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
"BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
        "certificate": "<certificate>",
        "environmentVariables": [
            {
                "name": "<environmentVariable-name>",
                "value": "<environmentVariable-value>",
                "type": "<environmentVariable-type>"
            }
        ],
        "registryCredential": [
            {
                "credential": "<credential-arn-or-name>",
                "credentialProvider": "<credential-provider>"
            }
        ],
        "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
        "privilegedMode": "<privileged-mode>"
    },
    "serviceRole": "<service-role>",
    "autoRetryLimit": <auto-retry-limit>,
    "timeoutInMinutes": <timeout>,
    "queuedTimeoutInMinutes": <queued-timeout>,
    "encryptionKey": "<encryption-key>",
    "tags": [
        {
            "key": "<tag-key>",
            "value": "<tag-value>"
        }
    ],
    "vpcConfig": {
        "securityGroupIds": [
            "<security-group-id>"
        ],
        "subnets": [
            "<subnet-id>"
        ]
    }
}

```

```

    ],
    "vpcId": "<vpc-id>"
  },
  "badgeEnabled": "<badge-enabled>",
  "logsConfig": {
    "cloudWatchLogs": {
      "status": "<cloudwatch-logs-status>",
      "groupName": "<group-name>",
      "streamName": "<stream-name>"
    },
    "s3Logs": {
      "status": "<s3-logs-status>",
      "location": "<s3-logs-location>",
      "encryptionDisabled": "<s3-logs-encryption-disabled>"
    }
  },
  "fileSystemLocations": [
    {
      "type": "EFS",
      "location": "<EFS-DNS-name-1>:/<directory-path>",
      "mountPoint": "<mount-point>",
      "identifier": "<efs-identifier>",
      "mountOptions": "<efs-mount-options>"
    }
  ],
  "buildBatchConfig": {
    "serviceRole": "<batch-service-role>",
    "combineArtifacts": <combine-artifacts>,
    "restrictions": {
      "maximumBuildsAllowed": <max-builds>,
      "computeTypesAllowed": [
        "<compute-type>"
      ],
      "fleetsAllowed": [
        "<fleet-name>"
      ]
    },
    "timeoutInMins": <batch-timeout>,
    "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
  },
  "concurrentBuildLimit": <concurrent-build-limit>
}

```

Substitua o seguinte:

name

Obrigatório. O nome deste projeto de build. Esse nome deve ser exclusivo em todos os projetos de compilação em sua AWS conta.

descrição

Opcional. A descrição desse projeto de build.

source

Obrigatório. Um [ProjectSource](#) objeto que contém informações sobre as configurações do código-fonte desse projeto de construção. Depois de adicionar um objeto source, você poderá adicionar até 12 outras origens com o . Essas configurações incluem o seguinte:

source/type

Obrigatório. O tipo de repositório que contém o código-fonte para build. Os valores válidos são:

- CODECOMMIT
- CODEPIPELINE
- GITHUB
- GITHUB_ENTERPRISE
- GITLAB
- GITLAB_SELF_MANAGED
- BITBUCKET
- S3
- NO_SOURCE

Se você usar NO_SOURCE, o buildspec não poderá ser um arquivo porque o projeto não terá uma origem. Em vez disso, você deve usar o atributo buildspec para especificar uma string formatada em YAML para seu buildspec. Para obter mais informações, consulte [Criar um projeto de compilação sem uma origem](#).

source/location

Obrigatório, a menos que você *<source-type>* defina comoCODEPIPELINE. O local do código-fonte para o tipo de repositório especificado.

- Para CodeCommit, o URL do clone HTTPS para o repositório que contém o código-fonte e o arquivo buildspec (por exemplo,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Para o Amazon S3, o nome do bucket de entrada da compilação, seguido pelo caminho e o nome do arquivo ZIP que contém o código-fonte e o buildspec. Por exemplo:
 - Para um arquivo ZIP localizado na raiz do bucket de entrada: `<bucket-name>/<object-name>.zip`.
 - Para um arquivo ZIP localizado em uma subpasta no bucket de entrada: `<bucket-name>/<subfolder-path>/<object-name>.zip`.
- Para GitHub, o URL do clone HTTPS para o repositório que contém o código-fonte e o arquivo buildspec. O URL deve conter github.com. Você deve conectar sua AWS conta à sua GitHub conta. Para fazer isso, use o CodeBuild console para criar um projeto de construção.
 - Escolha Autorizar aplicativo. (Depois de se conectar à sua GitHub conta, você não precisa concluir a criação do projeto de construção. Você pode fechar o CodeBuild console.)
- Para o GitHub Enterprise Server, a URL do clone HTTP ou HTTPS para o repositório que contém o código-fonte e o arquivo buildspec. Você também deve conectar sua AWS conta à sua conta do GitHub Enterprise Server. Para fazer isso, use o CodeBuild console para criar um projeto de construção.
 1. Crie um token de acesso pessoal no GitHub Enterprise Server.
 2. Copie esse token na sua área de transferência para que você possa usá-lo ao criar seu CodeBuild projeto. Para obter mais informações, consulte [Criação de um token de acesso pessoal para a linha de comando](#) no site da GitHub Ajuda.
 3. Ao usar o console para criar seu CodeBuild projeto, em Source, para Source provider, escolha GitHubEnterprise.
 4. Em Personal Access Token, cole o token que foi copiado na área de transferência. Escolha Salvar token. Sua CodeBuild conta agora está conectada à sua conta do GitHub Enterprise Server.
- Para GitLab e GitLab autogerenciado, o URL do clone HTTPS para o repositório que contém o código-fonte e o arquivo buildspec. Observe que, se você usar GitLab, o URL deverá conter gitlab.com. Se você usa GitLab autogerenciamento, o URL não precisa conter gitlab.com. Você deve conectar sua AWS conta à sua conta GitLab ou à sua conta GitLab autogerenciada. Para isso, use o console do CodeBuild para criar um projeto de compilação.

- No painel de navegação Developer Tools, escolha Configurações, Conexões e Criar conexão. Nessa página, crie uma conexão GitLab ou uma conexão GitLab autogerenciada e escolha Conectar a. GitLab
- Para o Bitbucket, o URL do clone HTTPS para o repositório que contém o código-fonte e o arquivo buildspec. O URL deve conter bitbucket.org. Você também deve conectar sua AWS conta à sua conta do Bitbucket. Para fazer isso, use o CodeBuild console para criar um projeto de construção.
 1. Quando você usa o console para se conectar (ou se reconectar) ao Bitbucket, na página Confirm access to your account do Bitbucket, escolha Grant access. (Depois de se conectar à sua conta do Bitbucket, você não precisa terminar de criar o projeto de compilação. Você pode fechar o CodeBuild console.)
- Para AWS CodePipeline, não especifique um location valor para source. CodePipeline ignora esse valor porque, ao criar um pipeline em CodePipeline, você especifica a localização do código-fonte no estágio Fonte do pipeline.

fonte/ gitCloneDepth

Opcional. A profundidade do histórico a ser obtido por download. O valor mínimo é 0. Se o valor for 0, superior a 25 ou não fornecido, o histórico completo será obtido por download com cada projeto de compilação. Se o tipo de origem é o Amazon S3, esse valor não é compatível.

source/buildspec

Opcional. A definição de especificação do build ou arquivo a ser usado. Se esse valor não for fornecido ou for definido como uma string vazia, o código-fonte deverá conter um arquivo buildspec.yml em seu diretório raiz. Se estiver definido, esse valor poderá ser uma definição de buildspec em linha, o caminho para um arquivo buildspec alternativo relativo ao diretório raiz da origem principal ou ao caminho para um bucket do S3. O bucket deve estar na mesma AWS região do projeto de construção. Especifique o arquivo buildspec usando seu ARN (por exemplo, arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml). Para obter mais informações, consulte [Nome do arquivo buildspec e local de armazenamento](#).

source/auth

Contém informações sobre as configurações de autorização CodeBuild para acessar o código-fonte a ser criado.

source/auth/type

Obrigatório. O tipo de autorização a ser usado. Os valores válidos são:

- OAUTH
- CODECONNECTIONS
- SECRETS_MANAGER

source/auth/resource

Opcional. O valor do recurso que se aplica ao tipo de autorização especificado. Isso pode ser o Secrets Manager ARN ou o CodeConnections ARN.

fonte/ reportBuildStatus

Especifica se é necessário enviar os status inicial e final de uma compilação ao seu provedor de origem. Se você definir isso com um provedor de origem diferente do GitHub GitHub Enterprise Server ou do Bitbucket, um `invalidInputException` será lançado.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

fonte/ buildStatusConfig

Contém informações que definem como o projeto de CodeBuild compilação reporta o status da compilação ao provedor de origem. Essa opção só é usada quando o provedor de origem é GITHUB, GITHUB_ENTERPRISE ou BITBUCKET.

fonte//contexto buildStatusConfig

Para origens do Bitbucket, esse parâmetro é usado para o parâmetro name no status de confirmação do Bitbucket. Para GitHub fontes, esse parâmetro é usado para o context parâmetro no status de GitHub confirmação.

Por exemplo, você pode fazer com que context contenha o número da compilação e o gatilho do webhook usando as variáveis de CodeBuild ambiente:

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -  
${CODEBUILD_WEBHOOK_TRIGGER}
```

Isso faz com que o contexto apareça assim para a compilação #24 acionada por um evento de solicitação pull de webhook:

```
AWS CodeBuild sample-project Build #24 - pr/8
```

fonte///URL buildStatusConfig de destino

Para origens do Bitbucket, esse parâmetro é usado para o parâmetro `url` no status de confirmação do Bitbucket. Para GitHub fontes, esse parâmetro é usado para o `target_url` parâmetro no status de GitHub confirmação.

Por exemplo, é possível definir o `targetUrl` como `https://aws.amazon.com/codebuild/<path to build>` e o status de confirmação será vinculado a esse URL.

Você também pode incluir variáveis de CodeBuild ambiente no `targetUrl` para adicionar informações adicionais ao URL. Por exemplo, para adicionar a região de compilação ao URL, defina o `targetUrl` como:

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Se a região de construção for `us-east-2`, ela se expandirá para:

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

fonte/ gitSubmodulesConfig

Opcional. Informações sobre a configuração dos submódulos do Git. Usado somente com CodeCommit GitHub,, GitHub Enterprise Server e Bitbucket.

fonte///fetchSubmodules gitSubmodulesConfig

Defina `fetchSubmodules` como `true` se quiser incluir os submódulos do Git em seu repositório. Os submódulos do Git incluídos devem ser configurados como HTTPS.

fonte/ InsecureSsl

Opcional. Usado somente com o GitHub Enterprise Server. Defina esse valor `true` para ignorar os avisos de TLS ao se conectar ao seu repositório de projetos do GitHub Enterprise Server. O valor padrão é `false`. `InsecureSsl` deve ser usado apenas para fins de teste. Ele não deve ser usado em um ambiente de produção.

source/sourceldentifier

Um identificador definido pelo usuário para a origem do projeto. Opcional para a origem principal. Obrigatório para origens secundárias.

secondarySources

Opcional. Uma matriz de [ProjectSource](#) objetos que contém informações sobre as fontes secundárias de um projeto de construção. É possível adicionar até 12 origens secundárias. Os objetos `secondarySources` usam as mesmas propriedades usadas pelo objeto `source`. Em um objeto de origem secundário, o `sourceIdentifier` é obrigatório.

secondarySourceVersions

Opcional. Uma matriz de objetos [ProjectSourceVersion](#). Se `secondarySourceVersions` for especificado no nível de compilação, ele terá precedência.

sourceVersion

Opcional. A versão da entrada de compilação a ser criada para esse projeto. Se não for especificada, a versão mais recente será usada. Se for especificada, ela deverá ser uma das seguintes:

- Para CodeCommit, o ID do commit, branch ou tag Git a ser usado.
- Para GitHub, o ID do commit, o ID do pull request, o nome da ramificação ou o nome da tag que corresponde à versão do código-fonte que você deseja criar. Caso seja especificado, um ID de solicitação deve usar o formato `pr/pull-request-ID` (por exemplo, `pr/25`). Se um nome de ramificação for especificado, o ID de confirmação HEAD da ramificação será usado. Se não estiver especificado, o ID de confirmação HEAD da ramificação padrão será usado.
- Para GitLab, o ID do commit, o ID do pull request, o nome da ramificação, o nome da tag ou a referência e um ID do commit. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).
- Para o Bitbucket, o ID de confirmação, o nome da ramificação ou o nome da tag que corresponda à versão do código-fonte que você deseja compilar. Se um nome de ramificação for especificado, o ID de confirmação HEAD da ramificação será usado. Se não estiver especificado, o ID de confirmação HEAD da ramificação padrão será usado.
- Para o Amazon S3, o ID da versão do objeto que representa o arquivo ZIP de entrada da compilação a ser usado.

Se `sourceVersion` for especificado no nível de compilação, essa versão terá precedência sobre esta `sourceVersion` (no nível do projeto). Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

artifacts

Obrigatório. Um [ProjectArtifacts](#) objeto que contém informações sobre as configurações do artefato de saída desse projeto de construção. Depois de adicionar um objeto `artifacts`, você poderá adicionar até 12 outros artefatos com o `.` Essas configurações incluem o seguinte:

artifacts/type

Obrigatório. O tipo de artefato de saída de build. Os valores válidos são:

- `CODEPIPELINE`
- `NO_ARTIFACTS`
- `S3`

artifacts/location

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

O nome do bucket de saída que você criou ou identificou nos pré-requisitos.

artifacts/path

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

O caminho do bucket de saída para colocar a pasta ou o arquivo ZIP. Se você não especificar um valor para `path`, CodeBuild usa `namespaceType` (se especificado) e `name` para determinar o caminho e o nome do arquivo ZIP ou pasta de saída da compilação. Por exemplo, se você especificar `MyPath` para `path`, `MyArtifact.zip` para `name`, o caminho e o nome serão `MyPath/MyArtifact.zip`.

artifacts/namespaceType

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

O namespace da pasta ou do arquivo ZIP de saída da compilação. Os valores válidos são `BUILD_ID` e `NONE`. Use `BUILD_ID` para inserir a ID de build no caminho da pasta ou do arquivo ZIP de saída do build. Caso contrário, use `NONE`. Se você não especificar um valor para `namespaceType`, CodeBuild usa `path` (se especificado) e `name` para determinar o caminho e o nome do arquivo ZIP ou pasta de saída da compilação. Por exemplo, se você especificar `MyPath` para `path`, `BUILD_ID` para `namespaceType` e `MyArtifact.zip` para `name`, o caminho e o nome serão `MyPath/build-ID/MyArtifact.zip`.

artifacts/name

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

O nome da pasta ou do arquivo ZIP de saída da compilação dentro de `location`. Por exemplo, se você especificar `MyPath` para `path`, `MyArtifact.zip` para `name`, o caminho e o nome serão `MyPath/MyArtifact.zip`.

artefatos/ overrideArtifactName

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

Opcional. Se definido como `true`, o nome especificado no bloco `artifacts` do arquivo `buildspec` substitui `name`. Para obter mais informações, consulte [Referência de especificação de construção para CodeBuild](#).

artifacts/packaging

Usado somente com o tipo de artefato do S3. Não é usado para outros tipos de artefatos.

Opcional. Especifica como empacotar os artefatos. Os valores permitidos são:

NONE

Crie uma pasta que contenha os artefatos de compilação. Este é o valor padrão.

ZIP

Crie um arquivo ZIP que contenha os artefatos de compilação.

secondaryArtifacts

Opcional. Uma matriz de [ProjectArtifacts](#) objetos que contém informações sobre as configurações de artefatos secundários para um projeto de construção. Você pode adicionar até 12 artefatos secundários. O `secondaryArtifacts` usa muitas das mesmas configurações usadas pelo objeto .

cache

Obrigatório. Um [ProjectCache](#) objeto que contém informações sobre as configurações de cache desse projeto de construção. Para obter mais informações, consulte [Compilações em cache](#).

ambiente

Obrigatório. Um [ProjectEnvironment](#) objeto que contém informações sobre as configurações do ambiente de construção desse projeto. Essas configurações incluem:

environment/type

Obrigatório. O tipo de ambiente de build. Para obter mais informações, consulte [digite](#) na Referência CodeBuild da API.

environment/image

Obrigatório. O identificador de imagem de Docker usado por esse ambiente de compilação. Normalmente, esse identificador é expresso como *image-name:tag*. Por exemplo, no repositório Docker CodeBuild usado para gerenciar suas imagens do Docker, isso pode ser. `aws/codebuild/standard:5.0` No Docker Hub, `maven:3.3.9-jdk-8`. No Amazon ECR, *account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag*. Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#).

environment/computeType

Obrigatório. Especifica os recursos computacionais usados por esse ambiente de compilação. Para obter mais informações, consulte [ComputeType](#) na CodeBuild Referência da API.

environment/certificate

Opcional. O ARN do bucket do Amazon S3, prefixo do caminho e chave de objeto que contém o certificado codificado em PEM. A chave de objeto pode ser apenas o arquivo .pem ou um arquivo .zip que contenha o certificado codificado em PEM. Por exemplo, se o nome do bucket do Amazon S3 for *<my-bucket>*, o prefixo do caminho for *<cert>* e o nome da chave do objeto for *<certificate.pem>*, os formatos aceitáveis para certificate serão *<my-bucket/cert/certificate.pem>* ou `arn:aws:s3:::<my-bucket/cert/certificate.pem>`.

environment/environmentVariables

Opcional. Uma matriz de [EnvironmentVariable](#) objetos que contém as variáveis de ambiente que você deseja especificar para esse ambiente de construção. Cada variável de ambiente é expressa como um objeto que contém um name, value e type de name, value e type.

O console e AWS CLI os usuários podem ver todas as variáveis de ambiente. Caso você não tenha problema em relação à visibilidade da variável de ambiente, defina os campos name e value e definir type como PLAINTEXT.

Recomendamos que você armazene variáveis de ambiente com valores confidenciais, como um ID de chave de AWS acesso, uma chave de acesso AWS secreta ou uma senha, como um parâmetro no Amazon EC2 Systems Manager Parameter Store ou AWS Secrets Manager. Para name esse parâmetro armazenado, defina um identificador CodeBuild para referência.

Se você usa o Amazon EC2 Systems Manager Parameter Store, para, defina o nome do parâmetro conforme armazenado no Parameter Store. Defina type como `PARAMETER_STORE`. Usando um parâmetro denominado `/CodeBuild/dockerLoginPassword` como exemplo, defina name como `LOGIN_PASSWORD`. Defina value como `/CodeBuild/dockerLoginPassword`. Defina type como `PARAMETER_STORE`.

Important

Se você usa o Amazon EC2 Systems Manager Parameter Store, recomendamos que você armazene parâmetros com nomes de parâmetros que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Você pode usar o CodeBuild console para criar um parâmetro no Amazon EC2 Systems Manager. Selecione **Create parameter** (Criar parâmetro) e siga as instruções na caixa de diálogo. (Nessa caixa de diálogo, para a chave KMS, você pode especificar o ARN de AWS KMS uma chave na sua conta. O Amazon EC2 Systems Manager usa essa chave para criptografar o valor do parâmetro durante o armazenamento e descriptografá-lo durante a recuperação.) Se você usar o CodeBuild console para criar um parâmetro, o console iniciará o nome do parâmetro com a `/CodeBuild/` forma como ele está sendo armazenado. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store, a função de serviço do projeto de construção deve permitir a `ssm:GetParameters` ação. Se você escolheu Nova função de serviço anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store com nomes de parâmetros que não começam com `/CodeBuild/`, e você escolheu Nova função de serviço, você deve atualizar essa função de serviço para permitir acesso aos nomes de parâmetros que não começam com `/CodeBuild/`. Isso porque essa função de serviço permite acesso apenas a nomes de parâmetro que comecem com `/CodeBuild/`.

Se você escolher Nova função de serviço, a função de serviço incluirá permissão para descriptografar todos os parâmetros sob o `/CodeBuild/` namespace no Amazon Systems EC2 Manager Parameter Store.

As variáveis de ambiente definidas por você substituem variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você definir uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já contiver uma variável de ambiente chamada `PATH` com um valor de `/usr/local/sbin:/usr/local/bin` e você definir uma variável de ambiente chamada `PATH` com um valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`.

Não defina nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome é definida em vários locais, o valor será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade.
- O valor na definição de projeto de build tem a precedência seguinte.
- O valor na declaração de buildspec tem a menor prioridade.

Se você usar o Secrets Manager, para `value`, defina o nome do parâmetro conforme armazenado no Secrets Manager. Defina `type` como `SECRETS_MANAGER`. Usando um segredo denominado `/CodeBuild/dockerLoginPassword` como exemplo, defina `name` como `LOGIN_PASSWORD`. Defina `value` como `/CodeBuild/dockerLoginPassword`. Defina `type` como `SECRETS_MANAGER`.

Important

Se você usa o Secrets Manager, recomendamos armazenar segredos com nomes que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

Se o projeto de compilação se referir a segredos armazenados no Secrets Manager, o perfil de serviço do projeto de compilação deverá permitir a ação `secretsmanager:GetSecretValue`. Se você escolheu Nova função de serviço anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se o projeto de compilação fizer referência a segredos armazenados no Secrets Manager com nomes que não comecem com `/CodeBuild/` e você selecionar Novo perfil de serviço, será necessário atualizar esse perfil de serviço para conceder acesso a nomes de segredo que não comecem com `/CodeBuild/`. O motivo é que o perfil de serviço permite acesso apenas a nomes de segredo que comecem com `/CodeBuild/`. Se você selecionar Novo perfil de serviço, o perfil de serviço incluirá permissão para descriptografar todos os segredos no namespace `/CodeBuild/` no Secrets Manager.

`environment/registryCredential`

Opcional. Um [RegistryCredential](#) objeto que especifica as credenciais que fornecem acesso a um registro privado do Docker.

`environment/registryCredential/credential`

Especifica o ARN ou o nome das credenciais criadas usando o AWS Managed Services. Você poderá usar o nome das credenciais somente se elas existirem na região atual.

`environment/registryCredential/credentialProvider`

O único valor válido é `SECRETS_MANAGER`.

Quando isso for configurado:

- `imagePullCredentials` deve ser definido como `SERVICE_ROLE`.
- A imagem não pode ser selecionada nem uma imagem do Amazon ECR.

`ambiente/tipo imagePullCredentials`

Opcional. O tipo de credencial CodeBuild usado para extrair imagens em sua compilação. Há dois valores válidos:

`CODEBUILD`

`CODEBUILD` Especifica que CodeBuild usa suas próprias credenciais. Você deve editar sua política de repositório do Amazon ECR para confiar no responsável pelo CodeBuild serviço.

`SERVICE_ROLE`

Especifica que CodeBuild usa a função de serviço do seu projeto de compilação.

Quando você usa uma conta ou imagem de registro privado, você deve usar credenciais `SERVICE_ROLE`. Ao usar uma imagem com CodeBuild curadoria, você deve usar `CODEBUILD` credenciais.

environment/privilegedMode

Defina `true` somente se planeja usar esse projeto de compilação para criar imagens do Docker. Do contrário, todas as compilações associadas que tentarem interagir com o daemon do Docker falharão. Você também deve iniciar o daemon do Docker, de maneira que as compilações possam interagir com ele. Uma maneira de fazer isso é inicializar o daemon do Docker na fase `install` de seu arquivo `buildspec` executando os seguintes comandos de compilação. Não execute esses comandos caso você tenha especificado uma imagem de ambiente da compilação fornecida pelo CodeBuild com suporte do Docker.

Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilégio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

serviceRole

Obrigatório. O ARN da função de serviço é CodeBuild usado para interagir com os serviços em nome do usuário (por exemplo, `arn:aws:iam::account-id:role/role-name`).

autoRetryLimit

Opcional. O número de repetições automáticas adicionais após a falha de uma compilação. Por exemplo, se o limite de repetição automática estiver definido como 2, CodeBuild chamará a `RetryBuild` API para repetir automaticamente sua compilação por até 2 vezes adicionais.

timeoutInMinutes

Opcional. O número de minutos, entre 5 e 2160 (36 horas), após os quais CodeBuild interrompe a construção se ela não for concluída. Se não especificado, o padrão de 60 será usado. Para determinar se e quando uma compilação CodeBuild foi interrompida devido a um tempo limite, execute o `batch-get-builds` comando. Para saber se o build parou, observe a saída para um

valor `buildStatus` de `FAILED`. Para saber qual foi o tempo limite do build, observe a saída para o valor `endTime` associado ao valor `phaseStatus` de `TIMED_OUT`.

`queuedTimeoutInMinutos`

Opcional. O número de minutos, entre 5 e 480 (8 horas), após os quais CodeBuild interrompe a compilação se ela ainda estiver na fila. Se não especificado, o padrão de 60 será usado.

`encryptionKey`

Opcional. O alias ou ARN do usado CodeBuild por para criptografar AWS KMS key a saída da compilação. Se você especificar um alias, use o formato `arn:aws:kms:region-ID:account-ID:key/key-ID` ou, se já houver um alias, use o formato `alias/key-alias`. Se não for especificada, a chave KMS AWS gerenciada para o Amazon S3 será usada.

`tags`

Opcional. Uma matriz de objetos [Tag](#) que fornecem as tags que você deseja associar a esse projeto de compilação. Você pode especificar até 50 tags. Essas tags podem ser usadas por qualquer AWS serviço que ofereça suporte à CodeBuild criação de tags de projeto. Cada tag é expressa como um objeto com uma `key` e um `value`.

`vpcConfig`

Opcional. Um [VpcConfig](#) objeto que contém informações sobre a configuração da VPC para seu projeto. Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

Essas propriedades incluem:

`vpcl`

Obrigatório. O ID da VPC que CodeBuild usa. Execute este comando para obter uma lista de todas as VPCs IDs em sua região:

```
aws ec2 describe-vpcs --region <region-ID>
```

`sub-redes`

Obrigatório. Uma matriz de sub-rede IDs que inclui recursos usados pelo CodeBuild. Execute este comando para obter estes IDs:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

securityGroupIds

Obrigatório. Uma matriz de grupos de segurança IDs usada por CodeBuild para permitir o acesso aos recursos na VPC. Execute este comando para obter estes IDs:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

badgeEnabled

Opcional. Especifica se você deseja incluir emblemas de construção em seu CodeBuild projeto. Defina como `true` para habilitar a criação de emblemas. Caso contrário, como `false`. Para obter mais informações, consulte [Crie uma amostra de crachás com CodeBuild](#).

logsConfig

Um [LogsConfig](#) objeto que contém informações sobre onde os registros dessa compilação estão localizados.

Configuração de logs/ cloudWatchLogs

Um [CloudWatchLogsConfig](#) objeto que contém informações sobre como enviar registros para o CloudWatch Logs.

logsConfig/s3Logs

Um LogsConfig objeto [do S3](#) que contém informações sobre o envio de registros para o Amazon S3.

fileSystemLocations

Opcional. Uma matriz de [ProjectFileSystemsLocation](#) objetos que contém informações sobre a configuração do Amazon EFS.

buildBatchConfig

Opcional. O `buildBatchConfig` objeto é uma [ProjectBuildBatchConfig](#) estrutura que contém as informações de configuração de compilação em lote para o projeto.

buildBatchConfig/ServiceRole

O ARN do perfil de serviço para o projeto de compilação em lote.

buildBatchConfig/Combine artefatos

Um valor booleano que especifica se os artefatos da compilação em lote devem ser combinados em um único local de artefato.

buildBatchConfig/restrições/ maximumBuildsAllowed

O número máximo de compilações permitidas.

buildBatchConfig/restrições/ computeTypesAllowed

Uma matriz de strings que especificam os tipos de computação que são permitidos para a compilação em lote. Consulte [Build environment compute types](#) para obter esses valores.

buildBatchConfig/restrições/ Frotas permitidas

Uma matriz de cadeias de caracteres que especificam as frotas permitidas para a construção em lote. Consulte [Executar construções em frotas de capacidade reservada](#) para obter mais informações.

buildBatchConfig/timeoutInMinutes

A quantidade máxima de tempo, em minutos, em que a compilação em lote deve ser concluída.

buildBatchConfig/batchReportMode

Especifica como os relatórios de status de compilação são enviados ao provedor de origem para a compilação em lote. Os valores válidos são:

REPORT_AGGREGATED_BATCH

(Padrão) Agregue todos os status de compilação em um único relatório de status.

REPORT_INDIVIDUAL_BUILDS

Envie um relatório de status separado para cada compilação individual.

concurrentBuildLimit

Define o número máximo de compilações simultâneas permitidas para este projeto.

Novas compilações só são iniciadas se o número atual de compilações for menor ou igual a esse limite. Se a contagem de compilações atual atender a esse limite, novas compilações serão limitadas e não serão executadas.

Criar o projeto

Para criar o projeto, execute o comando [create-project](#) novamente, transmitindo o arquivo JSON:

```
aws codebuild create-project --cli-input-json file://<json-file>
```

Se for bem-sucedida, a representação JSON de um objeto do [projeto](#) aparecerá na saída do console. Consulte a [sintaxe de CreateProject resposta](#) para ver um exemplo desses dados.

Exceto o nome de projeto de build, você pode alterar qualquer configuração do projeto de build, posteriormente. Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(AWS CLI\)](#).

Para começar a executar um build, consulte [Executar uma compilação \(AWS CLI\)](#).

Se seu código-fonte estiver armazenado em um GitHub repositório e você quiser CodeBuild reconstruir o código-fonte toda vez que uma alteração de código for enviada para o repositório, consulte [Começar a executar compilações automaticamente \(AWS CLI\)](#).

Criar um projeto de compilação (AWS SDKs)

Para obter informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Criar um projeto de compilação (AWS CloudFormation)

Para obter informações sobre como usar AWS CodeBuild com AWS CloudFormation, consulte [o AWS CloudFormation modelo CodeBuild](#) no Guia do AWS CloudFormation usuário.

Criar uma regra de notificação

Você pode usar regras de notificação para notificar os usuários quando ocorrem alterações importantes, como êxitos e falhas de compilação. As regras de notificação especificam os eventos e

o tópico do Amazon SNS utilizado para enviar notificações. Para obter mais informações, consulte [O que são notificações?](#)

Você pode usar o console ou o AWS CLI para criar regras de notificação para AWS CodeBuild.

Como criar uma regra de notificação (console)

1. Faça login no AWS Management Console e abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Selecione Build (Compilação), escolha Build projects (Projetos de compilação) e escolha um projeto de compilação em que você queira adicionar notificações.
3. Na página do projeto de compilação, escolha Notify (Notificar) e Create notification rule (Criar regra de notificação). Você também poderá acessar a página Settings (Configurações) do projeto de compilação e escolher 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 CodeBuild 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 projetos de compilação](#).
7. Em Targets (Destinos), siga um destes procedimentos:
 - Se você já configurou um recurso para usar com notificações, em Escolher tipo de destino, escolha Amazon Q Developer em aplicativos de bate-papo (Slack) ou tópico do SNS. Em Escolher destino, escolha o nome do cliente (para um cliente Slack configurado no Amazon Q Developer em aplicativos de bate-papo) ou o Amazon Resource Name (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 aplicativos de bate-papo como parte da criação de uma regra de notificação. Se você escolher o Amazon Q Developer em aplicativos de bate-papo (Slack), verá um botão orientando você a configurar um cliente no Amazon Q Developer em aplicativos de bate-papo. A escolha dessa opção abre o Amazon Q Developer no console de aplicativos de bate-papo. Para obter mais informações, consulte [Configurar integrações entre notificações e o Amazon Q Developer em aplicativos de bate-papo](#).
- Se você quiser usar um tópico existente do Amazon SNS como alvo, você deve adicionar a política necessária para AWS CodeStar Notificações, 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](#). Você também pode configurar a integração entre as notificações e o Amazon Q Developer em aplicativos de bate-papo para enviar notificações às salas de bate-papo do Amazon Chime. Para obter mais informações, consulte [Configurar a integração entre notificações e o Amazon Q Developer em aplicativos de bate-papo](#).

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 codestarnotifications 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*.

- 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 projeto de construção nomeado *MyBuildProject* em uma AWS conta com o ID *123456789012*. As notificações são enviadas com o tipo de detalhe completo para um tópico do Amazon SNS chamado *codestar-notifications-MyNotificationTopic* quando as compilações são bem-sucedidas:

```
{
  "Name": "MyNotificationRule",
  "EventTypeIds": [
    "codebuild-project-build-state-succeeded"
  ],
  "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",
  "Targets": [
    {
      "TargetType": "SNS",
      "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
    }
  ],
  "Status": "ENABLED",
  "DetailType": "FULL"
}
```

Salve o arquivo.

- 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 codestarnotifications create-notification-rule --cli-input-json
file://rule.json
```

- 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 as configurações do projeto de construção em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para alterar as configurações de um projeto de compilação.

Se você adicionar relatórios de teste a um projeto de compilação, verifique se o perfil do IAM tem as permissões descritas em [Permissões de relatório de teste](#).

Tópicos

- [Alterar configurações de um projeto de compilação \(console\)](#)
- [Alterar configurações de um projeto de compilação \(AWS CLI\)](#)
- [Alterar as configurações de um projeto de construção \(AWS SDKs\)](#)

Alterar configurações de um projeto de compilação (console)

Para alterar as configurações de um projeto de compilação, execute o seguinte procedimento:

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Execute um destes procedimentos:
 - Selecione o link para o projeto de compilação que você deseja alterar e selecione Build details (Detalhes da compilação).
 - Selecione o botão ao lado do projeto de compilação que você deseja alterar, selecione View details (Visualizar detalhes) e, em seguida, selecione Build details (Detalhes da compilação).

É possível modificar as seguintes seções:

Seções

- [Configuração de projetos](#)
- [Origem](#)
- [Environment](#)

- [Buildspec](#)
- [Configuração em lote](#)
- [Artefatos](#)
- [Logs](#)

Configuração de projetos

Na seção Configuração do projeto, selecione Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as propriedades a seguir.

Descrição

Insira uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.

Emblema da compilação

Selecione Habilitar emblema da compilação para que o status de compilação do projeto fique visível e possa ser incorporado. Para obter mais informações, consulte [Exemplo de emblemas de compilação](#).

Note

O emblema de compilação não se aplicará se o provedor de origem for Amazon S3.

Habilitar limite de compilações simultâneas

Se você quiser limitar o número de compilações simultâneas para esse projeto, execute as seguintes etapas:

1. Selecione Restringir número de compilações simultâneas que este projeto pode iniciar.
2. Em Limite de compilações simultâneas, insira o número máximo de compilações simultâneas permitidas para esse projeto. Esse limite não pode ser maior que o limite de compilações simultâneas definido para a conta. Se você tentar inserir um número maior que o limite da conta, uma mensagem de erro será exibida.

Novas compilações só são iniciadas se o número atual de compilações for menor ou igual a esse limite. Se a contagem de compilações atual atender a esse limite, novas compilações serão limitadas e não serão executadas.

Habilitar acesso a compilações públicas

Para disponibilizar os resultados de compilação do seu projeto ao público, incluindo usuários sem acesso a uma AWS conta, selecione Habilitar acesso público à compilação e confirme que você deseja tornar públicos os resultados da compilação. As seguintes propriedades são usadas para projetos de compilações públicas:

Perfil de serviço de compilação pública

Selecione Nova função de serviço se quiser CodeBuild criar uma nova função de serviço para você ou Função de serviço existente se quiser usar uma função de serviço existente.

A função pública do serviço de compilação CodeBuild permite ler os CloudWatch registros e baixar os artefatos do Amazon S3 para as compilações do projeto. Isso é necessário para disponibilizar os logs e os artefatos de compilação do projeto ao público.

Perfil de serviço

Insira o nome do novo perfil de serviço ou de um perfil de serviço existente.

Para tornar os resultados de compilação do projeto privados, desmarque Habilitar acesso a compilações públicas.

Para obter mais informações, consulte [Obtenha um projeto de construção público URLs](#).

Warning

Lembre-se do seguinte ao tornar públicos os resultados de compilação do projeto:

- Todos os resultados de compilação, os logs e os artefatos de um projeto, inclusive compilações executadas quando o projeto era privado, ficam disponíveis ao público.
- Todos os logs e os artefatos de compilação ficam disponíveis ao público. Variáveis de ambiente, código-fonte e outras informações confidenciais podem ter sido enviados aos logs e aos artefatos de compilação. É preciso ter cuidado com as informações enviadas aos logs de compilação. Algumas das práticas recomendadas são:
 - Não armazene valores confidenciais, especialmente chaves de AWS acesso IDs e chaves de acesso secretas, em variáveis de ambiente. Recomendamos que você use um repositório de parâmetros do Amazon EC2 Systems Manager ou AWS Secrets Manager armazene valores confidenciais.

- Siga [Práticas recomendadas para usar webhooks](#) para limitar quais entidades podem acionar uma compilação e não armazene o buildspec no próprio projeto, para garantir que os webhooks sejam o mais seguros possível.
- Um usuário mal-intencionado pode usar compilações públicas para distribuir artefatos danosos. Recomendamos que os administradores do projeto analisem todas as solicitações pull para verificar se a solicitação pull é uma alteração legítima. Também recomendamos que você valide todos os artefatos com as somas de verificação para garantir que os artefatos corretos estejam sendo baixados.

Informações adicionais

Em Tags, insira o nome e o valor de todas as tags que você deseja que AWS os serviços de suporte usem. Use Adicionar linha para adicionar uma tag. É possível adicionar até 50 tags.

Origem

Na seção Origem, escolha Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as seguintes propriedades:

Provedor de origem

Escolha o tipo de provedor do código-fonte. Use as seguintes listas para fazer seleções adequadas ao provedor de origem:

Note

CodeBuild não é compatível com o Bitbucket Server.

Amazon S3

Bucket

Escolha o nome do bucket de entrada que contém o código-fonte.

Chave de objeto do S3 ou pasta do S3

Insira o nome do arquivo ZIP ou o caminho na pasta que contém o código-fonte. Insira uma barra (/) para fazer download de tudo no bucket do S3.

Versão de origem

Insira o ID da versão do objeto que representa a compilação do arquivo de entrada. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

CodeCommit

Repositório

Selecione o repositório que deseja usar.

Tipo de referência

Selecione Ramificação, Tag do Git ou ID de confirmação para especificar a versão do código-fonte. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Bitbucket

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha CodeConnections, OAuth, Senha do aplicativo ou token de acesso pessoal para se conectar CodeBuild.

Conexão

Selecione uma conexão do Bitbucket ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório na minha conta Bitbucket ou repositório público e insira o URL do repositório.

Versão de origem

Insira uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para ter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#)

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o parâmetro name no status de confirmação do Bitbucket. Para obter mais informações, consulte [compilação](#) na documentação da API do Bitbucket.

Em URL de destino, insira o valor a ser usado para o parâmetro url no status de confirmação do Bitbucket. Para obter mais informações, consulte [compilação](#) na documentação da API do Bitbucket.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [Filtrar eventos de webhook do Bitbucket](#).

GitHub

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha GitHub Aplicativo ou Token de acesso pessoal para se conectar CodeBuild. OAuth

Conexão

Selecione uma GitHub conexão ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório em minha GitHub conta, repositório público ou webhook com GitHub escopo definido e insira a URL do repositório.

Versão de origem

Insira uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para ter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#)

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o context parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

Em URL de destino, insira o valor a ser usado para o target_url parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [GitHub eventos de webhook](#).

GitHub Enterprise Server

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

Escolha CodeConnectionso token de acesso pessoal ao qual se conectar CodeBuild.

Conexão

Selecione uma conexão GitHub Enterprise ou um segredo do Secrets Manager para se conectar por meio do tipo de conexão especificado.

Repositório

Escolha Repositório na minha conta GitHub corporativa ou webhook com escopo GitHub corporativo e insira a URL do repositório.

Versão de origem

Insira uma solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Submódulos do Git

Selecione Use Git submodules (Usar submódulos Git) se quiser incluir submódulos Git em seu repositório.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Em Contexto de status, insira o valor a ser usado para o `context` parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

Em URL de destino, insira o valor a ser usado para o `target_url` parâmetro no status de GitHub confirmação. Para obter mais informações, consulte [Criar um status de confirmação](#) no guia do GitHub desenvolvedor.

O status de uma compilação acionada por um webhook sempre é relatado ao provedor de origem. Para que o status de uma compilação iniciada no console ou de uma chamada de API seja comunicado ao provedor de origem, é necessário selecionar essa configuração.

Se as compilações do projeto forem acionadas por um webhook, será necessário enviar uma nova confirmação ao repositório para que uma alteração nessa configuração entre em vigor.

SSL inseguro

Selecione Habilitar SSL inseguro para ignorar os avisos de SSL ao se conectar ao seu GitHub repositório de projetos corporativos.

Em Eventos de webhook de origem primária, selecione Reconstruir sempre que uma alteração de código for enviada para esse repositório se você quiser CodeBuild criar o código-fonte toda vez que uma alteração de código for enviada para esse repositório. Para obter mais informações sobre webhooks e grupos de filtros, consulte [GitHub eventos de webhook](#).

GitLab

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

CodeConnections é usado para se GitLab conectar CodeBuild a.

Conexão

Selecione uma GitLab conexão pela qual se conectar CodeConnections.

Repositório

Selecione o repositório que deseja usar.

Versão de origem

Insira um ID de solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

GitLab Self Managed

Credencial

Escolha Credencial de origem padrão ou Credencial de origem personalizada e siga as instruções para gerenciar a credencial de origem padrão ou personalizar a credencial de origem.

Tipo de conexão

CodeConnectionsé usado para conectar o GitLab Self Managed CodeBuild a.

Conexão

Selecione uma conexão GitLab autogerenciada para se conectar CodeConnections.

Repositório

Selecione o repositório que deseja usar.

Versão de origem

Insira um ID de solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

Profundidade do clone do Git

Escolha Git clone depth (Profundidade de clone Git) para criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.

Status da compilação

Selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Environment

Na seção Ambiente, selecione Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as seguintes propriedades:

Modelo de provisionamento

Para alterar o modelo de provisionamento, escolha Alterar modelo de provisionamento e faça o seguinte:

- Para usar frotas sob demanda gerenciadas por AWS CodeBuild, escolha Sob demanda. Com frotas sob demanda, CodeBuild fornece computação para suas construções. As máquinas são destruídas quando a compilação termina. As frotas sob demanda são totalmente gerenciadas e incluem recursos de escalabilidade automática para lidar com picos de demanda.
- Para usar frotas de capacidade reservada gerenciadas por AWS CodeBuild, escolha Capacidade reservada e, em seguida, selecione o nome da frota. Com frotas de capacidade reservada, você configura um conjunto de instâncias dedicadas para seu ambiente de compilação. Essas máquinas permanecem ociosas, prontas para processar compilações ou testes imediatamente e reduzem a duração da compilação. Com frotas de capacidade reservada, suas máquinas estão sempre funcionando e continuarão a incorrer em custos enquanto forem provisionadas.

Para ter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#).

Imagem do ambiente

Para alterar a imagem de compilação, escolha Substituir imagem e faça o seguinte:

- Para usar uma imagem do Docker gerenciada por AWS CodeBuild, escolha Imagem gerenciada e, em seguida, faça seleções em Sistema operacional, Tempo de execução (s), Imagem e Versão da imagem. Faça uma seleção em Environment type (Tipo de ambiente) se estiver disponível.
- Para usar outra imagem do Docker, selecione Custom image (Imagem personalizada). Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Se você selecionar Other registry (Outro registro), em External registry URL (URL de registro externo), insira o nome e a tag da imagem do Docker no Docker Hub usando o formato *docker repository/docker image name*. Se você escolher o Amazon ECR, use o repositório Amazon ECR e a imagem do Amazon ECR para escolher a imagem do Docker em sua conta. AWS
- Para usar uma imagem privada do Docker, selecione Imagem personalizada. Em Tipo de ambiente, selecione ARM, Linux, Linux GPU ou Windows. Em Image registry (Registro da imagem), selecione Other registry (Outro registro) e insira o ARN das credenciais da imagem privada do Docker. As credenciais devem ser criadas pelo Secrets Manager. Para obter mais

informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

 Note

CodeBuild substitui as imagens personalizadas do ENTRYPOINT Docker.

Perfil de serviço

Execute um destes procedimentos:

- Se você não tiver uma função CodeBuild de serviço, escolha Nova função de serviço. No campo Role name, digite o nome da nova função.
- Se você tiver uma função CodeBuild de serviço, escolha Função de serviço existente. Em ARN do perfil, escolha o perfil de serviço.

 Note

Ao usar o console para criar um projeto de compilação, você pode criar uma função de CodeBuild serviço ao mesmo tempo. Por padrão, a função funciona somente com esse projeto de build. Se você usar o console para associar essa função de serviço a outro projeto de compilação, a função será atualizada para funcionar com os outros projetos de compilação. Uma função de serviço pode funcionar com até 10 projetos de compilação.

Configuração adicional

Timeout (Tempo limite)

Especifique um valor, entre 5 minutos e 36 horas, após o qual CodeBuild interrompe a compilação se ela não for concluída. Se hours e minutes forem deixados em branco, o valor padrão de 60 minutos será usado.

Privilegiado

Selecione Ativar este sinalizador se quiser criar imagens do Docker ou desejar que suas compilações obtenham privilégios elevados. somente se você planeja usar este projeto de compilação para compilar imagens do Docker. Do contrário, todas as compilações associadas que tentarem interagir com o daemon do Docker falharão. Você também deve

iniciar o daemon do Docker, de maneira que as compilações possam interagir com ele. Uma maneira de fazer isso é inicializar o daemon do Docker na fase `install` de especificação da compilação executando os comandos de compilação a seguir. Não execute esses comandos se você escolher uma imagem do ambiente de compilação fornecida CodeBuild com o suporte do Docker.

 Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Se você quiser CodeBuild trabalhar com sua VPC:

- Para VPC, escolha a ID da VPC que usa. CodeBuild
- Para sub-redes VPC, escolha as sub-redes que incluem os recursos que usa. CodeBuild
- Para grupos de segurança da VPC, escolha os grupos de segurança CodeBuild usados para permitir o acesso aos recursos no VPCs

Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

Computação

Selecione uma das opções disponíveis.

Credencial de registro

Especifique uma credencial de registro quando o projeto estiver configurado com uma imagem de registro não privada.

 Note

Essa credencial só será utilizada se as imagens forem substituídas pelas de registros privados.

Variáveis de ambiente

Insira o nome e o valor e, depois, selecione o tipo de cada variável de ambiente a ser usada pelas compilações.

 Note

CodeBuild define automaticamente a variável de ambiente para sua AWS região. Defina as seguintes variáveis de ambiente se elas não tiverem sido adicionadas ao `buildspec.yml`:

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

O console e AWS CLI os usuários podem ver as variáveis de ambiente. Caso você não tenha problema em relação à visibilidade da variável de ambiente, defina os campos Name e Value e Type como Plaintext.

Recomendamos que você armazene uma variável de ambiente com um valor confidencial, como um ID de chave de AWS acesso, uma chave de acesso AWS secreta ou uma senha como parâmetro no Amazon EC2 Systems Manager Parameter Store ou AWS Secrets Manager.

Se você usa o Amazon EC2 Systems Manager Parameter Store, em Tipo, escolha Parâmetro. Em Nome, insira um identificador CodeBuild para referência. Em Value, insira o nome do parâmetro conforme armazenado no Amazon EC2 Systems Manager Parameter Store. Usando um parâmetro chamado `/CodeBuild/dockerLoginPassword` como exemplo, em Type (Tipo), escolha Parameter (Parâmetro). Em Nome, digite `LOGIN_PASSWORD`. Em Valor, insira `/CodeBuild/dockerLoginPassword`.

 Important

Se você usa o Amazon EC2 Systems Manager Parameter Store, recomendamos que você armazene parâmetros com nomes de parâmetros que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Você pode usar o CodeBuild console para criar um parâmetro no Amazon EC2 Systems Manager. Selecione **Create parameter** (Criar parâmetro) e siga as instruções na caixa de diálogo. (Nessa caixa de diálogo, para a chave KMS, você pode especificar o ARN de AWS KMS uma chave na sua conta. O Amazon EC2 Systems Manager usa essa chave para criptografar o valor do parâmetro durante o armazenamento e descriptografá-lo durante a recuperação.) Se você usar o CodeBuild console para criar um parâmetro, o console iniciará o nome do parâmetro com a `/CodeBuild/` forma como ele está sendo armazenado. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store, a função de serviço do projeto de construção deve permitir a `ssm:GetParameters` ação. Se você escolheu Nova função de serviço anteriormente, CodeBuild incluía essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store com nomes de parâmetros que não começam com `/CodeBuild/`, e você escolheu Nova função de serviço, você deve atualizar essa função de serviço para permitir acesso aos nomes de parâmetros que não começam com `/CodeBuild/`. Isso porque essa função de serviço permite acesso apenas a nomes de parâmetro que comecem com `/CodeBuild/`.

Se você escolher Nova função de serviço, a função de serviço incluirá permissão para descriptografar todos os parâmetros sob o `/CodeBuild/` namespace no Amazon Systems EC2 Manager Parameter Store.

As variáveis de ambiente definidas por você substituem variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você definir uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já

contiver uma variável de ambiente chamada PATH com um valor de `/usr/local/sbin:/usr/local/bin` e você definir uma variável de ambiente chamada PATH com um valor de `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`.

Não defina nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome é definida em vários locais, o valor será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade.
- O valor na definição de projeto de build tem a precedência seguinte.
- O valor na declaração de buildspec tem a menor prioridade.

Se você usar o Secrets Manager, em Tipo, escolha Secrets Manager. Em Nome, insira um identificador CodeBuild para referência. Em Value (Valor), insira um `reference-key` usando o padrão `secret-id:json-key:version-stage:version-id`. Para ter mais informações, consulte [Secrets Manager reference-key in the buildspec file](#).

Important

Se você usa o Secrets Manager, recomendamos armazenar segredos com nomes que comecem com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager .

Se o projeto de compilação se referir a segredos armazenados no Secrets Manager, o perfil de serviço do projeto de compilação deverá permitir a ação `secretsmanager:GetSecretValue`. Se você escolheu Nova função de serviço anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação. No entanto, se você tiver selecionado Existing service role (Função de serviço existente), deverá incluir essa ação na função de serviço separadamente.

Se o projeto de compilação fizer referência a segredos armazenados no Secrets Manager com nomes que não comecem com `/CodeBuild/` e você selecionar Novo perfil de serviço, será necessário atualizar esse perfil de serviço para conceder acesso a nomes de segredo que não comecem com `/CodeBuild/`. O motivo é que

o perfil de serviço permite acesso apenas a nomes de segredo que comecem com `/CodeBuild/`.

Se você selecionar Novo perfil de serviço, o perfil de serviço incluirá permissão para descriptografar todos os segredos no namespace `/CodeBuild/` no Secrets Manager.

Buildspec

Na seção Buildspec, escolha Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as seguintes propriedades:

Especificações de compilação

Execute um destes procedimentos:

- Se o seu código-fonte inclui um arquivo buildspec, escolha Use a buildspec file (Usar um arquivo buildspec). Por padrão, o CodeBuild procura um arquivo chamado `buildspec.yml` no diretório raiz do código-fonte. Se o arquivo buildspec usar um nome ou um local diferente, insira o caminho a partir da raiz de origem em Nome do buildspec (por exemplo, `buildspec-two.yml` ou `configuration/buildspec.yml`). Se o arquivo buildspec estiver em um bucket do S3, ele deverá estar na mesma região da AWS do projeto de compilação. Especifique o arquivo buildspec usando o ARN (por exemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Se o código-fonte não incluir um arquivo de especificação de compilação ou se você quiser executar comandos de compilação diferentes dos especificados para a fase `build` no arquivo `buildspec.yml`, no diretório raiz do código-fonte, escolha Insert build commands. Para Build commands (Comandos de compilação), insira os comandos que você quer executar na fase `build`. Para vários comandos, separe-os com `&&` (por exemplo, `mvn test && mvn package`). Para executar comandos em outras fases, ou se você tiver uma lista longa de comandos para a fase `build`, adicione um arquivo `buildspec.yml` ao diretório raiz do código-fonte, adicione os comandos ao arquivo e escolha Usar o buildspec.yml no diretório raiz do código-fonte.

Para obter mais informações, consulte [Referência de buildspec](#).

Configuração em lote

Na seção Configuração em lote, selecione Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração. Para obter mais informações, consulte [Executar compilações em lotes](#).

É possível modificar as seguintes propriedades:

Perfil de serviço em lote

Fornece o perfil de serviço para compilações em lote.

Escolha uma das seguintes opções:

- Se você não tiver um perfil de serviço em lote, selecione Novo perfil de serviço. Em Perfil de serviço, insira um nome para o novo perfil.
- Se você tiver um perfil de serviço em lote, selecione Perfil de serviço existente. Em Perfil de serviço, selecione o perfil de serviço.

As compilações em lote introduzem um novo perfil de segurança na configuração em lote. Essa nova função é necessária, pois CodeBuild você deve poder chamar as `RetryBuild` ações `StartBuild``StopBuild`, e em seu nome para executar compilações como parte de um lote. Os clientes devem usar um novo perfil, e não o mesmo perfil que usam na compilação, por dois motivos:

- Fornecer ao perfil de compilação as permissões `StartBuild`, `StopBuild` e `RetryBuild` que permitem a uma única compilação iniciar mais compilações por meio do `buildspec`.
- CodeBuild compilações em lote fornecem restrições que restringem o número de compilações e tipos de computação que podem ser usados para as compilações no lote. Se o perfil de compilação tiver essas permissões, será possível que as próprias compilações ignorem essas restrições.

Tipos de computação permitidos para lote

Selecione os tipos de computação permitidos para o lote. Selecione todas as opções aplicáveis.

Frotas permitidas por lote

Selecione as frotas permitidas para o lote. Selecione todas as opções aplicáveis.

Máximo de compilações permitidas em lote

Insira o número máximo de compilações permitidas no lote. Se um lote exceder esse limite, ele falhará.

Tempo limite do lote

Insira o tempo máximo para a conclusão da compilação em lote.

Combinar artefatos

Selecione Combinar todos os artefatos do lote em um único local para que todos os artefatos do lote sejam combinados em um único local.

Modo de relatório em lote

Selecione o modo de relatório de status de compilação desejado para compilações em lote.

Note

Esse campo só está disponível quando a fonte do projeto é Bitbucket ou GitHub Enterprise GitHub, e Relatar os status de compilação ao provedor de origem quando suas compilações começam e terminam é selecionado em Fonte.

Compilações agregadas

Selecione para que os status de todas as compilações do lote sejam combinados em um único relatório de status.

Compilações individuais

Selecione para que os status de todas as compilações no lote sejam relatados separadamente.

Artefatos

Na seção Artefatos, escolha Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as seguintes propriedades:

Tipo

Execute um destes procedimentos:

- Se você não quiser criar artefatos de saída de build, escolha No artifacts. É recomendável fazer isso caso esteja executando apenas testes de compilação ou queira enviar uma imagem do Docker a um repositório do Amazon ECR.

- Para armazenar a saída de compilação em um bucket do S3, escolha Amazon S3 e faça o seguinte:
 - Se você quiser usar o nome do projeto para a pasta ou arquivo ZIP de saída da compilação, deixe Name (Nome) em branco. Caso contrário, insira o nome. (Se quiser um arquivo ZIP como saída e quiser que esse arquivo tenha uma extensão de arquivo, não se esqueça de incluir a extensão depois do nome do arquivo.)
 - Selecione Enable semântico versioning (Habilitar versionamento semântico) se quiser que um nome especificado no arquivo buildspec substitua qualquer nome especificado no console. O nome em um arquivo buildspec é calculado no tempo de compilação e usa a linguagem de comandos do Shell. Por exemplo, você pode anexar uma data e hora ao nome do artefato para que ele seja sempre exclusivo. Os nomes de artefato exclusivos impedem que os artefatos sejam substituídos. Para obter mais informações, consulte [Sintaxe de buildspec](#).
- Para Bucket name, selecione o nome do bucket de saída.
- Se você tiver escolhido Insert build commands (Inserir comandos de compilação) antes neste procedimento, em Output files (Arquivos de saída), insira os locais dos arquivos da compilação que deseja incluir no arquivo ZIP de saída da compilação ou na pasta. Para vários locais, separe-os com uma vírgula (por exemplo, appspec.yml, target/my-app.jar). Para obter mais informações, consulte a descrição de files em [Sintaxe de buildspec](#).
- Se você não quiser que os artefatos de compilação sejam criptografados, selecione Remove artifacts encryption (Remover a criptografia dos artefatos).

Para cada conjunto secundário de artefatos desejado:

1. Em Artifact identifier (Identificador de artefato), insira um valor com menos de 128 caracteres e que contenha apenas caracteres alfanuméricos e sublinhados.
2. Escolha Add artifact (Adicionar artefato).
3. Siga as etapas anteriores para configurar seus artefatos secundários.
4. Escolha Save artifact (Salvar artefato).

Configuração adicional

Chave de criptografia

Execute um destes procedimentos:

- Para usar a Chave gerenciada pela AWS Amazon S3 em sua conta para criptografar os artefatos de saída da compilação, deixe a chave de criptografia em branco. Esse é o padrão.
- Para usar uma chave gerenciada pelo cliente para criptografar os artefatos de saída da compilação, em Chave de criptografia, insira o ARN da chave gerenciada pelo cliente. Use o formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo de cache

Em Cache type (Tipo de cache), selecione uma das seguintes opções:

- Se não quiser usar um cache, escolha No cache.
- Se você quiser um cache do Amazon S3, selecione Amazon S3 e faça o seguinte:
 - Em Bucket, escolha o nome do bucket do S3 onde o cache está armazenado.
 - (Opcional) Em Prefixo do caminho do cache, insira um prefixo de caminho do Amazon S3. O valor Cache path prefix (Prefixo do caminho do cache) é semelhante ao nome de um diretório. Ele permite que você armazene o cache em um bucket no mesmo diretório.

Important

Não acrescente uma barra (/) ao final do prefixo do caminho.

- Se você quiser usar um cache local, selecione Local e depois selecione um ou mais modos de cache local.

Note

O modo de cache de camada do Docker está disponível apenas para o Linux. Se optar por esse modo, o projeto deverá ser executado no modo privilegiado.

O uso do cache economiza um tempo de compilação considerável porque as partes reutilizáveis do ambiente de compilação são armazenadas no cache e usadas em diferentes compilações. Para obter informações sobre como especificar um cache no arquivo de especificação de compilação, consulte [Sintaxe de buildspec](#). Para obter mais informações sobre armazenamento em cache, consulte [Compilações em cache para melhorar o desempenho](#).

Logs

Na seção Logs, selecione Editar. Quando as alterações forem concluídas, selecione Atualizar configuração para salvar a nova configuração.

É possível modificar as seguintes propriedades:

Selecione os logs que deseja criar. Você pode criar Amazon CloudWatch Logs, Amazon S3 logs ou ambos.

CloudWatch

Se você quiser CloudWatch registros do Amazon Logs:

CloudWatch logs

Selecione CloudWatch logs (Logs do &CW;).

Group name

Insira o nome do seu grupo de CloudWatch logs do Amazon Logs.

Nome do fluxo

Insira o nome do seu stream de CloudWatch log do Amazon Logs.

S3

Se você quiser que o Amazon S3 registre:

Logs do S3

Selecione S3 logs (Logs do S3).

Bucket

Escolha o nome do bucket do S3 para os logs.

Prefixo do caminho

Insira o prefixo para seus logs.

Desabilitar a criptografia de logs do S3

Selecione se não quiser que os logs do S3 sejam criptografados.

Alterar configurações de um projeto de compilação (AWS CLI)

Para obter informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Para atualizar um CodeBuild projeto com o AWS CLI, você cria um arquivo JSON com as propriedades atualizadas e passa esse arquivo para o [update-project](#) comando. Todas as propriedades não contidas no arquivo de atualização permanecem inalteradas.

No arquivo JSON de atualização, somente a propriedade name e as propriedades modificadas são necessárias. A propriedade name identifica o projeto a ser modificado. Para qualquer estrutura modificada, os parâmetros necessários para essas estruturas também devem ser incluídos. Por exemplo, para modificar o ambiente do projeto, as propriedades environment/type e environment/computeType são necessárias. Veja um de exemplo que atualiza a imagem do ambiente:

```
{
  "name": "<project-name>",
  "environment": {
    "type": "LINUX_CONTAINER",
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:4.0"
  }
}
```

Se você precisar obter os valores das propriedades atuais de um projeto, use o comando [batch-get-projects](#) para obter as propriedades atuais do projeto que você está modificando e gravar a saída em um arquivo.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

O *project-info.json* arquivo contém uma matriz de projetos, portanto, não pode ser usado diretamente para atualizar um projeto. No entanto, você pode copiar as propriedades que deseja modificar do *project-info.json* arquivo e colá-las no arquivo de atualização como uma linha de base para as propriedades que deseja modificar. Para obter mais informações, consulte [Visualizar detalhes de um projeto de compilação \(AWS CLI\)](#).

Modifique o arquivo JSON de atualização conforme descrito em [Criar um projeto de compilação \(AWS CLI\)](#) e salve os resultados. Quando terminar de modificar o arquivo JSON de atualização, execute o comando [update-project](#), transmitindo o arquivo JSON de atualização.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

Se for bem-sucedido, o JSON do projeto atualizado será exibido na saída. Se algum parâmetro necessário estiver ausente, uma mensagem de erro será exibida na saída que identifica os parâmetros ausentes. Por exemplo, esta será a mensagem de erro exibida se o parâmetro `environment/type` estiver ausente:

```
aws codebuild update-project --cli-input-json file://update-project.json
```

```
Parameter validation failed:  
Missing required parameter in environment: "type"
```

Alterar as configurações de um projeto de construção (AWS SDKs)

Para obter informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Vários tokens de acesso em CodeBuild

CodeBuild suporta o fornecimento de tokens de acesso a provedores terceirizados a partir de seus segredos em AWS Secrets Manager ou por meio de Conexões de código da AWS conexões. Você pode definir seu segredo ou conexão como a credencial padrão para interações com um provedor terceirizado específico GitHub, como GitHub Enterprise ou Bitbucket.

Você pode definir suas credenciais de origem em três níveis diferentes:

1. Credenciais no nível da conta para todos os projetos: essas são as credenciais padrão para todos os projetos em uma conta da AWS . Elas serão usadas em um projeto quando nenhuma credencial de projeto ou de nível de origem for especificada.
2. Credenciais no nível da fonte para um repositório específico: é quando um segredo ou CodeConnections conexão do Secrets Manager é definido em uma fonte do projeto. Essas credenciais serão usadas somente para operações no repositório de origem especificado. Isso permite que você configure vários tokens de acesso com diferentes escopos de permissão no mesmo projeto e não use as credenciais padrão no nível da conta.
3. Credenciais de fallback no nível do projeto: você pode definir uma credencial de fallback no nível do projeto usando `NO_SOURCE` como tipo de fonte primária e definir um segredo ou uma conexão nela. Isso pode ser usado quando você tem várias fontes em um projeto, mas deseja usar as

mesmas credenciais ou quando não deseja usar as credenciais padrão no nível da conta para o projeto.

Tópicos

- [Etapa 1: Criar um segredo ou uma CodeConnections conexão do Secrets Manager](#)
- [Etapa 2: conceder acesso à função IAM do CodeBuild projeto aos segredos do Secrets Manager](#)
- [Etapa 3: Configurar o Secrets Manager ou CodeConnections tokens](#)
- [Opções de configuração adicionais](#)

Etapa 1: Criar um segredo ou uma CodeConnections conexão do Secrets Manager

Use as instruções a seguir para criar um segredo ou uma CodeConnections conexão do Secrets Manager:

- [Criar e armazenar um token em um segredo do Secrets Manager.](#)
- [Crie uma conexão com GitHub](#)
- [Crie uma conexão com o GitHub Enterprise Server](#)
- [Criar uma conexão com o Bitbucket](#)

Etapa 2: conceder acesso à função IAM do CodeBuild projeto aos segredos do Secrets Manager

Note

Antes de continuar, você deve ter acesso ao token criado no Secrets Manager ou CodeConnections.

Para conceder acesso ao papel do IAM do CodeBuild projeto ao Secrets Manager ou CodeConnections, você deve adicionar a seguinte política do IAM.

Para conceder acesso à função IAM do CodeBuild projeto

1. Crie uma função do IAM para seu CodeBuild projeto seguindo as instruções [CodeBuild Permitir interagir com outros AWS serviços](#) para seu CodeBuild projeto.
2. Execute um destes procedimentos:
 - Adicione a seguinte política do IAM ao papel CodeBuild do seu projeto para conceder acesso ao seu segredo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "<secret-arn>"
      ]
    }
  ]
}
```

(Opcional) Se você estiver usando chaves gerenciadas pelo AWS KMS cliente para criptografar um segredo do Secrets Manager, poderá adicionar a seguinte declaração de política para conceder acesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "<kms-key-arn>",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:SecretARN": "<secret-arn>"
        }
      }
    }
  ]
}
```



```
    }  
  ]  
}
```

- Adicione a seguinte política do IAM ao papel CodeBuild do seu projeto para conceder acesso à sua conexão.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "codeconnections:GetConnectionToken",  
        "codeconnections:GetConnection"  
      ],  
      "Resource": [  
        "<connection-arn>"  
      ]  
    }  
  ]  
}
```

Etapa 3: Configurar o Secrets Manager ou CodeConnections tokens

Você pode definir suas credenciais de origem em três níveis diferentes com o Secrets Manager ou CodeConnections tokens.

Configure o Secrets Manager ou CodeConnections tokens como credenciais no nível da conta

Você pode configurar um segredo ou uma CodeConnections conexão do Secrets Manager como uma credencial no nível da conta e usá-la em um projeto.

AWS Management Console

Para configurar uma conexão como uma credencial de nível de conta no AWS Management Console

1. Em Provedor de origem, escolha Bitbucket ou GitHub Enterprise. GitHub
2. Em Credencial, siga um destes procedimentos:

- Escolha Credencial de origem padrão para usar a credencial de origem padrão da conta e aplicá-la a todos os projetos.
 - a. Se você ainda não tiver feito login no provedor de origem, escolha Gerenciar credencial de origem padrão.
 - b. Em Tipo de credencial, escolha um tipo de credencial.
 - c. Se você escolher CodeConnections, escolha usar uma conexão existente ou criar uma nova conexão.

Se você escolheu outro tipo de credencial, em Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:

- Se você optar por usar o Secrets Manager, poderá usar uma conexão secreta existente ou criar um segredo e escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).
- Se você optar por usar CodeBuild, insira seu token ou seu nome de usuário e senha do aplicativo e escolha Salvar.
- Escolha Credencial de origem personalizada para usar uma credencial de origem personalizada para substituir as configurações padrão da conta.
 - a. Em Tipo de credencial, escolha um tipo de credencial.
 - b. Em Conexão, escolha usar uma conexão existente ou crie uma.

AWS CLI

Para configurar uma conexão como uma credencial de nível de conta no AWS CLI

- Abra um terminal (Linux, macOS ou Unix) ou um prompt de comando (Windows). Use o AWS CLI para executar o `import-source-credentials` comando.

Use o seguinte comando para configurar um segredo do Secrets Manager:

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

Use o comando a seguir para configurar uma CodeConnections conexão:

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

Esse comando permite importar um token como credenciais de origem padrão no nível da conta. Ao importar uma credencial usando a [ImportSourceCredentials](#) API, CodeBuild usará o token para todas as interações com o provedor de origem, como webhooks, relatórios de status de compilação e operações de clonagem do git, a menos que um conjunto mais específico de credenciais tenha sido configurado no projeto.

Agora você pode usar o token no projeto de compilação e executá-lo. Para ter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) e [Execute AWS CodeBuild compilações manualmente](#).

Configurar vários tokens como credenciais no nível de origem

Para usar segredos ou CodeConnections conexões do Secrets Manager como credenciais no nível da fonte, faça referência direta ao token no CodeBuild projeto e inicie uma compilação.

AWS Management Console

Para configurar vários tokens como credenciais de nível de origem no AWS Management Console

1. Em Source provider, escolha GitHub.
2. Em Credencial, siga um destes procedimentos:
 - Escolha Credencial de origem padrão para usar a credencial de origem padrão da conta e aplicá-la a todos os projetos.
 - a. Se você não estiver conectado GitHub, escolha Gerenciar credencial de origem padrão.
 - b. Em Tipo de credencial, escolha GitHub Aplicativo.
 - c. Em Conexão, escolha usar uma conexão existente ou crie uma.

- Escolha Credencial de origem personalizada para usar uma credencial de origem personalizada para substituir as configurações padrão da conta.
 - a. Em Tipo de credencial, escolha GitHub Aplicativo.
 - b. Em Conexão, escolha usar uma conexão existente ou crie uma.
- 3. Escolha Adicionar origem e repita o processo de escolha do provedor de origem e das credenciais.

AWS CLI

Para configurar vários tokens como credenciais de nível de origem no AWS CLI

- Abra um terminal (Linux, macOS ou Unix) ou um prompt de comando (Windows). Use o AWS CLI para executar o create-project comando.

Use o seguinte comando:

```
aws codebuild create-project --region <aws-region> \
  --name <project-name> \
  --artifacts type=NO_ARTIFACTS \
  --environment "type=LINUX_CONTAINER,
                 computeType=BUILD_GENERAL1_SMALL,
                 image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
  --service-role <service-role-name> \
  --source "type=GITHUB,
            location=<github-repository-1>,
            auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-1>}" \
  --secondary-sources "type=GITHUB,
                       location=<github-repository-2>,
                       auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-2>},
                       sourceIdentifier=secondary"

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

Defina um fallback de credencial de origem no nível do projeto

Para configurar o fallback da credencial de origem no nível do projeto, use NO_SOURCE como fonte primária do seu projeto e faça referência ao token.

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

Ao usar NO_SOURCE, um buildspec normalmente é fornecido dentro do modelo de origem, pois não está diretamente configurado para usar uma fonte externa para buscar [buildspec](#). Normalmente, uma origem NO_SOURCE lidará com a clonagem de todos os repositórios relevantes de dentro do buildspec. Para garantir que as credenciais configuradas estejam disponíveis para essas operações, você pode ativar a opção `git-credential-helper` no buildspec.

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

Durante a compilação, CodeBuild lerá o AuthServer campo do token configurado e usará as credenciais do token para todas as solicitações do git para esse provedor de origem terceirizado específico.

Opções de configuração adicionais

Você pode configurar as credenciais em nível de conta do Secrets Manager usando AWS CloudFormation modelos. Você pode usar o AWS CloudFormation modelo a seguir para definir uma credencial no nível da conta:

```
Parameters:  
  GitHubToken:  
    Type: String  
    NoEcho: true  
    Default: placeholder
```

```
Resources:
  CodeBuildAuthTokenSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token
      Name: codebuild-auth-token
      SecretString:
        !Join
        - ''
        - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
          - !Ref GitHubToken
          - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
          Value: personal_access_token
  CodeBuildSecretsManagerAccountCredential:
    Type: AWS::CodeBuild::SourceCredential
    Properties:
      ServerType: GITHUB
      AuthType: SECRETS_MANAGER
      Token: !Ref CodeBuildAuthTokenSecret
```

Note

Se você também estiver criando um projeto na mesma pilha, use o AWS CloudFormation atributo [DependsOn](#) para garantir que ele `AccountCredential` seja criado antes do projeto.

Você também pode configurar as credenciais de vários níveis de origem do Secrets Manager usando AWS CloudFormation modelos. Você pode usar o AWS CloudFormation modelo a seguir para usar vários tokens para extrair várias fontes:

```
Parameters:
  GitHubTokenOne:
    Type: String
    NoEcho: true
    Default: placeholder
  GitHubTokenTwo:
    Type: String
    NoEcho: true
```

Default: placeholder

Resources:

CodeBuildSecretsManagerProject:

Type: AWS::CodeBuild::Project

Properties:

Name: codebuild-multitoken-example

ServiceRole: *<service-role>*

Environment:

Type: LINUX_CONTAINER

ComputeType: BUILD_GENERAL1_SMALL

Image: aws/codebuild/amazonlinux-x86_64-standard:5.0

Source:

Type: GITHUB

Location: *<github-repository-one>*

Auth:

Type: SECRETS_MANAGER

Resource: !Ref CodeBuildAuthTokenSecretOne

SecondarySources:

- Type: GITHUB

Location: *<github-repository-two>*

Auth:

Type: SECRETS_MANAGER

Resource: !Ref CodeBuildAuthTokenSecretTwo

SourceIdentifier: secondary

Artifacts:

Type: NO_ARTIFACTS

LogsConfig:

CloudWatchLogs:

Status: ENABLED

CodeBuildProjectIAMRoleSecretAccess:

Type: AWS::IAM::RolePolicy

Properties:

RoleName: *<role-name>*

PolicyName: CodeBuildProjectIAMRoleSecretAccessPolicy

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- secretsmanager:GetSecretValue

Resource:

- !Ref CodeBuildAuthTokenSecretOne

- !Ref CodeBuildAuthTokenSecretTwo

```
CodeBuildAuthTokenSecretOne:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: CodeBuild auth token one
    Name: codebuild-auth-token-one
    SecretString:
      !Join
      - ''
      - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
        - !Ref GitHubTokenOne
        - '"}'
    Tags:
      - Key: codebuild:source:provider
        Value: github
      - Key: codebuild:source:type
        Value: personal_access_token
CodeBuildAuthTokenSecretTwo:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: CodeBuild auth token two
    Name: codebuild-auth-token-two
    SecretString:
      !Join
      - ''
      - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
        - !Ref GitHubTokenTwo
        - '"}'
    Tags:
      - Key: codebuild:source:provider
        Value: github
      - Key: codebuild:source:type
        Value: personal_access_token
```

Excluir projetos de compilação em AWS CodeBuild

Você pode usar o CodeBuild console, AWS CLI, ou AWS SDKs para excluir um projeto de compilação no CodeBuild. Se você excluir um projeto, suas compilações não serão excluídas.

 Warning

Não é possível excluir um projeto que tenha compilações e uma política de recursos. Para excluir um projeto com uma política de recursos e compilações, primeiro é necessário remover a política de recursos e excluir suas compilações.

Tópicos

- [Excluir um projeto de compilação \(console\)](#)
- [Excluir um projeto de compilação \(AWS CLI\)](#)
- [Excluir um projeto de compilação \(AWS SDKs\)](#)

Excluir um projeto de compilação (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Execute um destes procedimentos:
 - Escolha o botão de seleção próximo ao projeto de build a ser excluído, selecione Delete (Excluir).
 - Selecione o link do projeto de build a ser excluído e escolha Delete.

 Note

Por padrão, somente os 10 projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione um valor diferente para Projects per page (Projetos por página) ou use as setas voltar e avançar para visualizar os projetos.

Excluir um projeto de compilação (AWS CLI)

1. Execute o comando `delete-project`:

```
aws codebuild delete-project --name name
```

Substitua o seguinte espaço reservado:

- ***name***: Cadeia de caracteres obrigatória. O nome do projeto de build a ser excluído. Para uma lista de projetos de build disponíveis, execute o comando `list-projects`. Para obter mais informações, consulte [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#).

2. Se bem-sucedido, nenhum dado ou erro aparecerá na saída.

Para obter mais informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Excluir um projeto de compilação (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Obtenha um projeto de construção público URLs

AWS CodeBuild permite que você disponibilize os resultados de compilação, registros e artefatos de seus projetos de construção para o público em geral. Isso permite que os colaboradores dos repositórios de origem visualizem os resultados e baixem os artefatos de uma compilação, sem exigir que tenham acesso a uma conta da AWS .

Ao disponibilizar as compilações do projeto ao público, todos os resultados de compilação, os logs e os artefatos de um projeto, inclusive compilações executadas quando o projeto era privado, ficam disponíveis ao público. Da mesma forma, ao tornar privado um projeto de compilação público, os resultados de compilação desse projeto deixam de estar disponíveis ao público.

Para obter informações sobre como alterar a visibilidade pública dos resultados de compilação de um projeto, consulte [Habilitar acesso a compilações públicas](#).

CodeBuild fornece uma URL para as compilações públicas do seu projeto que é exclusiva do seu projeto.

Para obter o URL público para o projeto de compilação, siga este procedimento.

Para obter o URL de um projeto de compilação público

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. No painel de navegação, selecione Projetos de compilação.
3. Escolha o link para o projeto de compilação para o qual você deseja obter o URL público.
4. O URL público é exibido no campo URL público do projeto, na seção Configuração. É possível escolher o link para abrir o URL ou copiar o URL com o botão Copiar.

Warning

Lembre-se do seguinte ao tornar públicos os resultados de compilação do projeto:

- Todos os resultados de compilação, os logs e os artefatos de um projeto, inclusive compilações executadas quando o projeto era privado, ficam disponíveis ao público.
- Todos os logs e os artefatos de compilação ficam disponíveis ao público. Variáveis de ambiente, código-fonte e outras informações confidenciais podem ter sido enviados aos logs e aos artefatos de compilação. É preciso ter cuidado com as informações enviadas aos logs de compilação. Algumas das práticas recomendadas são:
 - Não armazene valores confidenciais, especialmente chaves de AWS acesso IDs e chaves de acesso secretas, em variáveis de ambiente. Recomendamos que você use um repositório de parâmetros do Amazon EC2 Systems Manager ou AWS Secrets Manager armazene valores confidenciais.
 - Siga [Práticas recomendadas para usar webhooks](#) para limitar quais entidades podem acionar uma compilação e não armazene o buildspec no próprio projeto, para garantir que os webhooks sejam o mais seguros possível.
- Um usuário mal-intencionado pode usar compilações públicas para distribuir artefatos danosos. Recomendamos que os administradores do projeto analisem todas as solicitações pull para verificar se a solicitação pull é uma alteração legítima. Também recomendamos que você valide todos os artefatos com as somas de verificação para garantir que os artefatos corretos estejam sendo baixados.

Compartilhar projetos de compilação

O compartilhamento de projetos permite que os proprietários compartilhem seus AWS CodeBuild projetos com outras AWS contas ou usuários. Neste modelo, a conta que possui o projeto (proprietário) compartilha um projeto com outras contas (consumidores). Um consumidor não pode editar nem executar um projeto.

Tópicos

- [Compartilhar um projeto](#)
- [Serviços relacionados](#)
- [Acesse CodeBuild projetos compartilhados com você](#)
- [Cancelar o compartilhamento de um projeto compartilhado](#)
- [Identificar um projeto compartilhado](#)
- [Permissões para projetos compartilhados](#)

Compartilhar um projeto

O consumidor pode usar o AWS CodeBuild console, o AWS CLI e o console para visualizar o projeto e as construções que você compartilhou. O consumidor não pode editar nem executar o projeto.

Você pode adicionar um projeto a um compartilhamento de recursos existente ou criar um no [console do AWS RAM](#).

Note

Não é possível excluir um projeto com compilações que foram adicionadas a um compartilhamento de recursos.

Para compartilhar um projeto com unidades organizacionais ou toda uma organização, é necessário habilitar o compartilhamento com AWS Organizations. Para obter mais informações, consulte [Habilitar o compartilhamento com o AWS Organizations](#) no Guia do usuário do AWS RAM .

Você pode usar o AWS CodeBuild console, o AWS RAM console ou o AWS CLI para compartilhar um projeto de sua propriedade.

Pré-requisitos para compartilhamento de projetos

Antes de começar a compartilhar um projeto, verifique se sua AWS conta é proprietária. Não é possível compartilhar um projeto que tenha sido compartilhado com você.

Para compartilhar um projeto que você possui (CodeBuild console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.

 Note

Por padrão, somente os 10 projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Projetos por página ou use as setas para voltar e avançar.

3. Escolha o projeto que deseja compartilhar e selecione Compartilhar. Para obter mais informações, consulte [Create a resource share](#) no Guia do usuário do AWS RAM .

Para compartilhar um projeto que você possui (AWS RAM console)

Consulte [Creating a resource share](#) no Guia do usuário do AWS RAM .

Para compartilhar um projeto que você possui (AWS RAM comando)

Use o comando [create-resource-share](#).

Como compartilhar um projeto de sua propriedade (comando do CodeBuild)

Use o comando [put-resource-policy](#):

1. Crie um arquivo chamado `policy.json` e copie o seguinte nele.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "<consumer-aws-account-id-or-user>"
    },
    "Action": [
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"
    ],
    "Resource": "<arn-of-project-to-share>"
  }]
}
```

2. Atualize `policy.json` com o ARN do projeto e os identificadores com quais compartilhá-lo. O exemplo a seguir concede acesso somente de leitura ao usuário raiz da AWS conta identificada por 123456789012.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    },
    "Action": [
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"
    ],
    "Resource": "arn:aws:codebuild:us-west-2:123456789012:project/my-project"
  ]
}
```

3. Execute o comando [put-resource-policy](#).

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Obtenha o ARN do compartilhamento de AWS RAM recursos.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Isso vai gerar uma resposta semelhante a esta:

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
      "lastUpdatedTime": "<last-update-time>"
    }
  ]
}
```

Na resposta, copie o *<resource-share-arn>* valor a ser usado na próxima etapa.

5. Execute o comando AWS RAM [promote-resource-share-created-from-policy](#).

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-share-arn>
```

Serviços relacionados

O compartilhamento de projetos se integra ao AWS Resource Access Manager (AWS RAM), um serviço que possibilita que você compartilhe seus AWS recursos com qualquer AWS conta ou por meio AWS Organizations de. Com o AWS RAM, você compartilha recursos criando um compartilhamento de recursos que especifica os recursos e os consumidores com os quais compartilhá-los. Os consumidores podem ser AWS contas individuais, unidades organizacionais ou uma organização inteira em AWS Organizations. AWS Organizations

Para obter mais informações, consulte o Guia do usuário do [AWS RAM](#).

Acesse CodeBuild projetos compartilhados com você

Para acessar um projeto compartilhado, o perfil do IAM de um consumidor requer a permissão `BatchGetProjects`. É possível anexar a seguinte política ao perfil do IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetProjects"
  ]
}
```

Para obter mais informações, consulte [Usando políticas baseadas em identidade para AWS CodeBuild](#).

Cancelar o compartilhamento de um projeto compartilhado

Um projeto não compartilhado, incluindo suas compilações, pode ser acessado somente por seu proprietário. Se você cancelar o compartilhamento de um projeto, qualquer AWS conta ou usuário com quem você o compartilhou anteriormente não poderá acessar o projeto ou suas compilações.

Para cancelar o compartilhamento de um projeto de sua propriedade, é necessário removê-lo do compartilhamento de recursos. Você pode usar o AWS CodeBuild AWS RAM console, o console ou AWS CLI fazer isso.

Para cancelar o compartilhamento de um projeto compartilhado de sua propriedade (AWS RAM console)

Consulte [Atualização de um compartilhamento de recursos](#) no Guia do usuário do AWS RAM .

Como cancelar o compartilhamento de um projeto compartilhado de sua propriedade (AWS CLI)

Use o comando [disassociate-resource-share](#).

Para cancelar o compartilhamento do projeto que você possui (CodeBuild comando)

Execute o [delete-resource-policy](#) comando e especifique o ARN do projeto que você deseja descompartilhar:

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Identificar um projeto compartilhado

Proprietários e consumidores podem usar o AWS CLI para identificar projetos compartilhados.

Para identificar projetos compartilhados com sua AWS conta ou usuário (AWS CLI)

Use o [list-shared-projects](#) comando para retornar os projetos que estão compartilhados com você.

Permissões para projetos compartilhados

Permissões para proprietários

Um proprietário do projeto pode editar o projeto e usá-lo para executar compilações.

Permissões para clientes

Um consumidor de projeto pode exibir um projeto e suas compilações, mas não pode editar um projeto ou usá-lo para executar compilações.

Projetos de compilação de tags

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. Cada AWS tag tem duas partes:

- Uma chave de tag (por exemplo `CostCenter`, `Environment`, `Project` ou `Secret`). Chaves de tag fazem distinção entre maiúsculas e 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. Como chaves de tag, os valores das tags diferenciam maiúsculas de minúsculas.

Juntos, esses são conhecidos como pares de chave-valor. Para obter informações sobre o número máximo de tags em um projeto e as restrições em relação a chaves e valores das tags, consulte [Tags](#).

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 CodeBuild projeto que você atribui a um bucket do S3. Para obter mais informações sobre o uso de tags, consulte [Tagging Best Practices](#).

Em CodeBuild, os recursos primários são o projeto e o grupo de relatórios. Você pode usar o CodeBuild console, o AWS CLI CodeBuild APIs, ou AWS SDKs para adicionar, gerenciar e remover tags de um projeto. Além de identificar, organizar e monitorar o projeto com as tags, é possível usá-las em políticas do IAM para ajudar a controlar quem pode visualizar e interagir com o projeto. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Important

Ao usar o recurso de capacidade reservada, os dados armazenados em cache nas instâncias da frota, incluindo arquivos de origem, camadas do Docker e diretórios em cache especificados no buildspec, podem ser acessados por outros projetos na mesma conta. Isso ocorre por design e permite que projetos dentro da mesma conta compartilhem instâncias de frota.

Tópicos

- [Adicionar uma tag a um projeto](#)
- [Visualizar tags de um projeto](#)
- [Editar tags de um projeto](#)
- [Remover uma tag de um projeto](#)

Adicionar uma tag a um projeto

Adicionar tags a um projeto pode ajudar você a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Primeiro, adicione uma ou mais tags (pares de chave/valor) a um projeto. Lembre-se de que há limites no número de tags que é possível ter em um projeto. Há restrições com relação aos caracteres que é possível usar nos campos chave e valor. Para obter mais informações, consulte [Tags](#). Depois que tiver tags, será possível criar políticas do IAM para gerenciar o acesso ao projeto com base nessas tags. Você pode usar o CodeBuild console ou o AWS CLI para adicionar tags a um projeto.

Important

Ao usar o recurso de capacidade reservada, os dados armazenados em cache nas instâncias da frota, incluindo arquivos de origem, camadas do Docker e diretórios em cache especificados no buildspec, podem ser acessados por outros projetos na mesma conta. Isso ocorre por design e permite que projetos dentro da mesma conta compartilhem instâncias de frota.

Para obter mais informações sobre como adicionar tags a um projeto ao criá-lo, consulte [Adicionar uma tag a um projeto \(console\)](#).

Important

Antes de adicionar uma tag a um projeto, revise as políticas do IAM que podem usar tags para controlar o acesso a recursos, como projetos de compilação. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Tópicos

- [Adicionar uma tag a um projeto \(console\)](#)

- [Adicionar uma tag a um projeto \(AWS CLI\)](#)

Adicionar uma tag a um projeto (console)

Você pode usar o CodeBuild console para adicionar uma ou mais tags a um CodeBuild projeto.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Projetos de compilação, escolha o nome do projeto ao qual deseja adicionar tags.
3. No painel de navegação, selecione Configurações. Escolha Tags do projeto de compilação.
4. Se nenhuma tag tiver sido adicionada ao projeto, escolha 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 projeto (AWS CLI)

Para adicionar uma tag a um projeto ao criá-lo, consulte [Criar um projeto de compilação \(AWS CLI\)](#). Em `create-project.json`, adicione as tags.

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](#).

Se houver êxito, o comando não retornará nada.

Visualizar tags de um projeto

As tags podem ajudar você a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Para obter mais informações sobre o uso de tags, consulte o informe [Práticas recomendadas de marcação](#). Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Visualizar tags de um projeto (console)

Você pode usar o CodeBuild console para visualizar as tags associadas a um CodeBuild projeto.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Projetos de compilação, escolha o nome do projeto no qual deseja visualizar tags.
3. No painel de navegação, selecione Configurações. Escolha Tags do projeto de compilação.

Visualizar tags de um projeto (AWS CLI)

Para visualizar tags de um projeto de compilação, execute o comando a seguir. Use o nome do projeto para o parâmetro `--names`.

```
aws codebuild batch-get-projects --names your-project-name
```

Se tiver êxito, esse comando retornará informações formatadas em JSON sobre o projeto de compilação que incluem algo como o seguinte:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "JanesProject"
  }
}
```

Se o projeto não tiver tags, a seção `tags` estará vazia:

```
"tags": []
```

Editar tags de um projeto

É possível alterar o valor de uma tag associada a um projeto. Também é possível alterar o nome da chave, o que é equivalente a excluir 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 obter mais informações, consulte [Tags](#).

Important

A edição de tags de um projeto pode afetar o acesso a ele. Antes de editar o nome (chave) ou o valor de uma tag de um projeto, revise as políticas do IAM que podem usar essa chave ou esse valor para uma tag a fim de controlar o acesso a recursos, como projetos de

compilação. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Editar uma tag de um projeto (console)

Você pode usar o CodeBuild console para editar as tags associadas a um CodeBuild projeto.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Projetos de compilação, escolha o nome do projeto do qual deseja editar tags.
3. No painel de navegação, selecione Configurações. Escolha Tags do projeto de compilação.
4. Selecione Editar.
5. 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 projeto (AWS CLI)

Para adicionar, alterar ou excluir tags de um projeto de compilação, consulte [Alterar configurações de um projeto de compilação \(AWS CLI\)](#). Atualize a seção tags nos dados formatados em JSON usados para atualizar o projeto.

Remover uma tag de um projeto

É possível remover uma ou mais tags associadas a um projeto. A remoção de uma tag não exclui a tag de outros AWS recursos associados a essa tag.

Important

A remoção de tags de um projeto pode afetar o acesso a ele. Antes de remover uma tag de um projeto, revise as políticas do IAM que podem usar a chave ou o valor para uma tag a fim de controlar o acesso a recursos, como projetos de compilação. Para obter exemplos

de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Remover uma tag de um projeto (console)

Você pode usar o CodeBuild console para remover a associação entre uma tag e um CodeBuild projeto.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Projetos de compilação, escolha o nome do projeto do qual deseja remover tags.
3. No painel de navegação, selecione Configurações. Escolha Tags do projeto de compilação.
4. Selecione Editar.
5. Encontre a tag que deseja excluir e selecione Remover tag.
6. Quando terminar de excluir tags, selecione Enviar.

Remover uma tag de um projeto (AWS CLI)

Para excluir uma ou mais tags de um projeto de compilação, consulte [Alterar configurações de um projeto de compilação \(AWS CLI\)](#). Atualize a seção tags nos dados formatados em JSON com uma lista atualizada de tags que não contenha as que você deseja excluir. Se quiser excluir todas as tags, atualize a seção tags para:

```
"tags: []"
```

Note

Se você excluir um projeto de CodeBuild compilação, todas as associações de tags serão removidas do projeto de compilação excluído. Não é necessário remover as tags antes de excluir um projeto de compilação.

Use corretores com AWS CodeBuild

AWS CodeBuild oferece suporte à integração com executores de GitHub ações, executores autogerenciados e o GitLab executor Buildkite.

Tópicos

- [Executores de GitHub ações auto-hospedados em AWS CodeBuild](#)
- [GitLab Corredores autogerenciados em AWS CodeBuild](#)
- [Executador autogerenciado do Buildkite em AWS CodeBuild](#)

Executores de GitHub ações auto-hospedados em AWS CodeBuild

Você pode configurar seu projeto para configurar executores de GitHub ações auto-hospedados em CodeBuild contêineres para processar seus trabalhos de fluxo de trabalho do GitHub Actions. Isso pode ser feito configurando um webhook usando seu CodeBuild projeto e atualizando seu fluxo de trabalho de GitHub ações YAML para usar executores auto-hospedados hospedados em máquinas. CodeBuild

As etapas de alto nível para configurar um CodeBuild projeto para executar trabalhos de GitHub ações são as seguintes:

1. Se você ainda não tiver feito isso, crie um token de acesso pessoal ou conecte-se a um OAuth aplicativo ao qual conectar seu projeto GitHub.
2. Navegue até o CodeBuild console, crie um CodeBuild projeto com um webhook e configure seus filtros de webhook.
3. Atualize seu fluxo de trabalho de GitHub ações YAML GitHub para configurar seu ambiente de compilação.

Para obter um procedimento mais detalhado, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

Esse recurso permite que seus trabalhos de fluxo de trabalho do GitHub Actions obtenham integração nativa AWS, o que fornece segurança e conveniência por meio de recursos como IAM AWS CloudTrail, AWS Secrets Manager integração e Amazon VPC. Você pode acessar os tipos de instância mais recentes, incluindo instâncias baseadas em ARM.

Tópicos

- [Sobre o executor CodeBuild de GitHub ações hospedado](#)
- [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#)
- [Solucionar problemas do webhook](#)
- [Substituições de rótulos suportadas pelo executor de ações CodeBuild hospedado GitHub](#)

- [Imagens de computação compatíveis com o executor CodeBuild de ações hospedado GitHub](#)

Sobre o executor CodeBuild de GitHub ações hospedado

A seguir estão algumas perguntas comuns sobre o executor CodeBuild de GitHub ações hospedado.

Quando devo incluir as substituições de imagem e instância no rótulo?

Você pode incluir as substituições de imagem e instância no rótulo para especificar um ambiente de criação diferente para cada um dos trabalhos do fluxo de trabalho do GitHub Actions. Isso pode ser feito sem a necessidade de criar vários CodeBuild projetos ou webhooks. Por exemplo, isso é útil quando você precisa usar uma [matriz para os trabalhos de fluxo de trabalho](#).

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      image:${{ matrix.os }}
      instance-size:${{ matrix.size }}
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

Note

As aspas podem ser necessárias se houver `runs-on` vários rótulos contendo o contexto de GitHub ações.

Posso usar AWS CloudFormation esse recurso?

Sim, você pode incluir um grupo de filtros em seu AWS CloudFormation modelo que especifica um filtro de eventos de trabalho do fluxo de trabalho do GitHub Actions no webhook do seu projeto.


```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Para obter mais informações, consulte [Filtrar eventos de GitHub webhook \(\)AWS CloudFormation](#).

Se precisar de ajuda para configurar as credenciais do projeto em seu AWS CloudFormation modelo, consulte [AWS::CodeBuild::SourceCredential](#)o Guia do AWS CloudFormation usuário para obter mais informações.

Como posso mascarar segredos ao usar esse recurso?

Por padrão, os segredos impressos no log não são mascarados. Se você quiser mascarar os segredos, poderá usar a seguinte sintaxe: `::add-mask::value`. Veja abaixo um exemplo de como você pode usar essa sintaxe no YAML:

```
name: Secret Job
on: [push]
jobs:
  Secret-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    env:
      SECRET_NAME: "secret-name"
    steps:
      - run: echo "::add-mask::$SECRET_NAME"
```

Para obter mais informações, consulte [Mascarar um valor em um logon](#) GitHub.

Posso receber eventos de webhook do GitHub Actions de vários repositórios em um único projeto?

CodeBuild oferece suporte a webhooks de nível organizacional e global, que recebem eventos de uma organização ou empresa específica. Para obter mais informações, consulte [GitHub webhooks globais e organizacionais](#).

Quais regiões oferecem suporte ao uso de um executor CodeBuild de GitHub ações hospedado?

CodeBuildOs executores de GitHub ações hospedados são suportados em todas as CodeBuild regiões. Para obter mais informações sobre Regiões da AWS onde CodeBuild está disponível, consulte [AWS Serviços por região](#).

Quais plataformas oferecem suporte ao uso de um executor CodeBuild de GitHub ações hospedado?

CodeBuildOs executores de GitHub ações hospedados são compatíveis tanto na Amazon EC2 quanto [AWS Lambda](#) na computação. É possível usar as seguintes plataformas: Amazon Linux 2, Amazon Linux 2023, Ubuntu e Windows Server Core 2019. Para obter mais informações, consulte [EC2 imagens de computação](#) e [Imagens de computação do Lambda](#).

Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado

Este tutorial mostra como configurar seus CodeBuild projetos para executar trabalhos do GitHub Actions. Para obter mais informações sobre o uso de GitHub ações com, CodeBuild consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

Para concluir este tutorial, você deve primeiramente:

- Conecte-se com um token de acesso pessoal, um segredo, OAuth aplicativo ou GitHub aplicativo do Secrets Manager. Se quiser se conectar a um OAuth aplicativo, você deve usar o CodeBuild console para fazer isso. Se quiser criar um token de acesso pessoal, você pode usar o CodeBuild console ou usar a [ImportSourceCredentials API](#). Para obter mais instruções, consulte [GitHub e acesso ao GitHub Enterprise Server em CodeBuild](#).
- Conecte-se CodeBuild à sua GitHub conta. Para fazer isso, é possível utilizar uma das seguintes opções:
 - Você pode adicionar GitHub como provedor de origem no console. Você pode se conectar com um token de acesso pessoal, um segredo do Secrets Manager, OAuth aplicativo ou GitHub aplicativo. Para instruções, consulte [GitHub e acesso ao GitHub Enterprise Server em CodeBuild](#).
 - Você pode importar suas GitHub credenciais por meio da [ImportSourceCredentials API](#). Isso só pode ser feito com um token de acesso pessoal. Se você se conectar usando um OAuth aplicativo, deverá se conectar usando o console em vez disso. Para instruções, consulte [Conecte-se GitHub com um token de acesso \(CLI\)](#).

Note

Isso só precisa ser feito se você não estiver conectado GitHub à sua conta.

Etapa 1: criar um CodeBuild projeto com um webhook

Nesta etapa, você criará um CodeBuild projeto com um webhook e o revisará no GitHub console. Você também pode escolher GitHub Enterprise como seu provedor de origem. Para saber mais sobre a criação de um webhook no GitHub Enterprise, consulte [GitHub webhooks manuais](#).

Para criar um CodeBuild projeto com um webhook

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
3. Em Tipo de projeto, escolha Projeto Runner.

Em Runner:

- a. Para o provedor Runner, escolha GitHub.
- b. Em Localização do executor, escolha Repositório.
- c. Para URL do repositório em Repositório, escolha `https://github.com/user-name/nome do repositório`.

Note

Por padrão, o projeto só receberá eventos `WORKFLOW_JOB_QUEUED` para um único repositório. Para receber eventos de todos os repositórios dentro de uma organização ou empresa, consulte [GitHub webhooks globais e organizacionais](#).

4.
 - Em Ambiente:
 - Escolha uma imagem de ambiente e computação compatíveis. Observe que você tem a opção de substituir as configurações de imagem e instância usando um rótulo no seu fluxo de trabalho de GitHub ações YAML. Para obter mais informações, consulte [Etapa 2: atualizar seu fluxo de trabalho de GitHub ações YAML](#).
 - Em Buildspec:
 - Observe que o buildspec será ignorado, a menos que `buildspec-override:true` seja adicionado como rótulo. Em vez disso, o CodeBuild substituirá para usar comandos que configurarão o executor auto-hospedado.
5. Continue com os valores padrão e escolha Criar projeto de compilação.

6. Abra o GitHub console em <https://github.com/user-name/repository-name/settings/hooks> para verificar se um webhook foi criado e está habilitado para fornecer eventos de trabalhos do Workflow.

Etapa 2: atualizar seu fluxo de trabalho de GitHub ações YAML

Nesta etapa, você atualizará o arquivo YAML do fluxo de trabalho do GitHub Actions [GitHub](#) para configurar seu ambiente de compilação e usar os executores auto-hospedados do GitHub Actions em. CodeBuild Para obter mais informações, consulte [Usar rótulos com executores auto-hospedados](#) e [Substituições de rótulos suportadas pelo executor de ações CodeBuild hospedado GitHub](#).

Atualize seu fluxo de trabalho de GitHub ações (YAML)

Navegue até [GitHub](#) atualize a [runs-on](#) configuração em seu fluxo de trabalho de GitHub ações YAML para configurar seu ambiente de compilação. Para fazer isso, é possível utilizar uma das seguintes opções:

- Você pode especificar o nome do projeto e o ID da execução. Nesse caso, a compilação usará a configuração do projeto existente para computação, imagem, versão da imagem e tamanho da instância. O nome do projeto é necessário para vincular as configurações AWS relacionadas do seu trabalho de GitHub ações a um CodeBuild projeto específico. Ao incluir o nome do projeto no YAML, CodeBuild é permitido invocar trabalhos com as configurações corretas do projeto. Ao fornecer o ID de execução, CodeBuild mapeará sua compilação para execuções de fluxo de trabalho específicas e interromperá a compilação quando a execução do fluxo de trabalho for cancelada. Para obter mais informações, consulte o [contexto do github](#).

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

Note

Certifique-se de que seu nome **<project-name>** corresponda ao nome do projeto que você criou na etapa anterior. Se não corresponder, não CodeBuild processará o webhook e o fluxo de trabalho de GitHub ações poderá travar.

Veja a seguir um exemplo de um fluxo de trabalho de GitHub ações YAML:

```
name: Hello World
```

```
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    steps:
      - run: echo "Hello World!"
```

- Você também pode substituir a imagem e o tipo de computação no rótulo. Consulte [Imagens de computação compatíveis com o executor CodeBuild de ações hospedado GitHub](#) para obter uma lista de imagens selecionadas. Para usar imagens personalizadas, consulte [Substituições de rótulos suportadas pelo executor de ações CodeBuild hospedado GitHub](#). O tipo de computação e a imagem no rótulo substituirão as configurações do ambiente no projeto. Para substituir suas configurações de ambiente para uma compilação de computação CodeBuild EC2 ou Lambda, use a seguinte sintaxe:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    image:<environment-type>-<image-identifier>
    instance-size:<instance-size>
```

Veja a seguir um exemplo de um fluxo de trabalho de GitHub ações YAML:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:arm-3.0
        instance-size:small
    steps:
      - run: echo "Hello World!"
```

- Você pode substituir a frota usada para a compilação no rótulo. Isso substituirá as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#). Para substituir as configurações da sua frota para uma compilação EC2 computacional da Amazon, use a seguinte sintaxe:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

```
fleet:<fleet-name>
```

Para substituir a frota e a imagem usadas para a compilação, use a seguinte sintaxe:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
    image:<environment-type>-<image-identifier>
```

Veja a seguir um exemplo de um fluxo de trabalho de GitHub ações YAML:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        fleet:myFleet
        image:arm-3.0
    steps:
      - run: echo "Hello World!"
```

- Para executar seus trabalhos de GitHub ações em uma imagem personalizada, você pode configurar uma imagem personalizada em seu CodeBuild projeto e evitar fornecer um rótulo de substituição de imagem. CodeBuild usará a imagem configurada no projeto se nenhum rótulo de substituição de imagem for fornecido.
- Opcionalmente, você pode fornecer rótulos fora dos CodeBuild compatíveis. Esses rótulos serão ignorados com o objetivo de substituir os atributos da compilação, mas não falharão na solicitação do webhook. Por exemplo, adicionar `testLabel` como rótulo não impedirá a execução da compilação.

Note

Se uma dependência fornecida por GitHub -hosted runners não estiver disponível no CodeBuild ambiente, você poderá instalar a dependência usando GitHub Ações na execução do fluxo de trabalho. Por exemplo, você pode usar a ação [setup-python](#) para instalar o Python no ambiente de compilação.

Execute os comandos buildspec nas fases INSTALL, PRE_BUILD e POST_BUILD

Por padrão, CodeBuild ignora qualquer comando buildspec ao executar uma compilação de ações auto-hospedada. Para executar comandos buildspec durante a compilação, `buildspec-override:true` pode ser adicionado como sufixo ao rótulo:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    buildspec-override:true
```

Ao usar esse comando, CodeBuild criará uma pasta chamada `actions-runner` na pasta de origem primária do contêiner. Quando o executor de GitHub ações for iniciado durante a BUILD fase, o executor será executado no `actions-runner` diretório.

Há várias limitações ao usar uma substituição de buildspec em uma compilação de ações auto-hospedada: GitHub

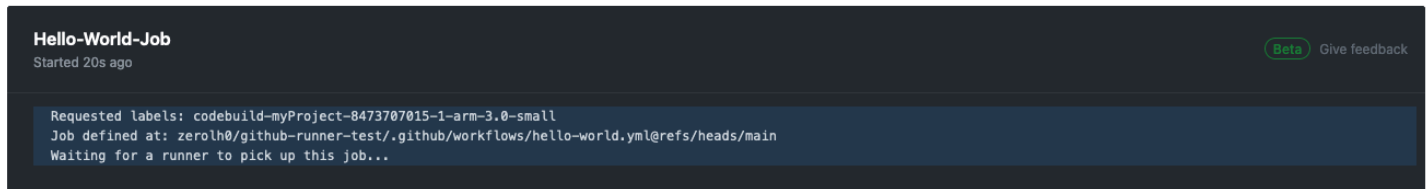
- CodeBuild não executará comandos buildspec durante a BUILD fase, pois o executor auto-hospedado é executado na fase. BUILD
- CodeBuild não baixará nenhuma fonte primária ou secundária durante a DOWNLOAD_SOURCE fase. Se você tiver um arquivo buildspec configurado, somente esse arquivo será baixado da fonte primária do projeto.
- Se um comando de compilação falhar na INSTALL fase PRE_BUILD ou, o executor auto-hospedado não CodeBuild será iniciado e o trabalho do fluxo de trabalho de GitHub ações precisará ser cancelado manualmente.
- CodeBuild busca o token do executor durante a DOWNLOAD_SOURCE fase, que tem um tempo de expiração de uma hora. Se suas INSTALL fases PRE_BUILD ou fases excederem uma hora, o token do corredor poderá expirar antes do início do corredor GitHub auto-hospedado.

Etapa 3: revisar os resultados

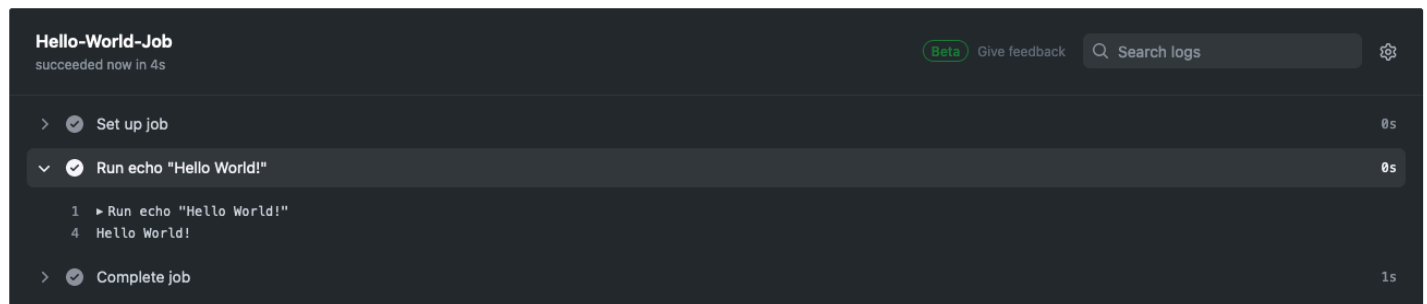
Sempre que ocorre a execução de um fluxo de trabalho do GitHub Actions, CodeBuild receberia os eventos do trabalho do fluxo de trabalho por meio do webhook. Para cada trabalho no fluxo de trabalho, CodeBuild inicia uma compilação para executar um executor de ações efêmero GitHub. O executor é responsável por executar um único trabalho de fluxo de trabalho. Depois que o trabalho for concluído, o executor e o processo de compilação associado serão encerrados imediatamente.

Para visualizar os registros de trabalhos do seu fluxo de trabalho, navegue até seu repositório GitHub, escolha Ações, escolha o fluxo de trabalho desejado e, em seguida, escolha o trabalho específico do qual você gostaria de revisar os registros.

Você pode revisar os rótulos solicitados no registro enquanto o trabalho está esperando para ser realizado por um executor auto-hospedado. CodeBuild



Depois que o trabalho for concluído, você poderá visualizar o log do trabalho.



GitHub Opções de configuração do executor de ações

Você pode especificar as seguintes variáveis de ambiente na configuração do seu projeto para modificar a configuração de seus executores auto-hospedados.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

CodeBuild registrará executores auto-hospedados no nome da organização especificado como o valor dessa variável de ambiente. Para obter mais informações sobre o registro de corredores no nível da organização e as permissões necessárias, consulte [Criar configuração para um just-in-time corredor de uma](#) organização.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

CodeBuild registrará executores auto-hospedados no nome da empresa especificado como o valor dessa variável de ambiente. Para obter mais informações sobre o registro de executores no nível corporativo e as permissões necessárias, consulte [Criar configuração para um just-in-time executor para uma](#) empresa.

 Note

Os executores corporativos não estão disponíveis para os repositórios da organização por padrão. Para que executores auto-hospedados realizem trabalhos de fluxo de trabalho, talvez seja necessário definir suas configurações de acesso ao grupo de executores. Para obter mais informações, consulte [Disponibilizando enterprise runners para repositórios](#).

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

CodeBuild registrará os executores auto-hospedados no ID inteiro do grupo de executores armazenado como o valor dessa variável de ambiente. Por padrão, esse valor é 1. Para obter mais informações sobre grupos de executores auto-hospedados, consulte [Gerenciando o acesso a executores auto-hospedados usando](#) grupos.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

Para configurar o registro de executores no nível da organização usando o arquivo YAML do fluxo de trabalho do GitHub Actions, você pode usar a seguinte sintaxe:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        organization-registration-name:myOrganization
    steps:
      - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

Para configurar o registro de executores de nível corporativo usando o arquivo YAML do fluxo de trabalho do GitHub Actions, você pode usar a seguinte sintaxe:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
```

```
runs-on:
  - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    enterprise-registration-name:myEnterprise
steps:
  - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

Para configurar o registro de executores em um ID de grupo de executores específico usando o arquivo YAML do fluxo de trabalho do GitHub Actions, você pode usar a seguinte sintaxe:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        registration-group-id:3
    steps:
      - run: echo "Hello World!"
```

Filtrar eventos de webhook de GitHub ações ()AWS CloudFormation

A seguinte parte formatada em YAML de um modelo do AWS CloudFormation cria um grupo de filtros que aciona uma compilação quando ela é avaliada como verdadeira. O grupo de filtros a seguir especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions com um nome de fluxo de trabalho correspondente à expressão `\[CI-CodeBuild\]` regular.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
```

```

    Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
  Triggers:
    Webhook: true
    ScopeConfiguration:
      Name: organization-name
      Scope: GITHUB_ORGANIZATION
    FilterGroups:
      - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
      - Type: WORKFLOW_NAME
        Pattern: \[CI-CodeBuild\]

```

Filtrar eventos de webhook de GitHub ações (AWS CDK)

O AWS CDK modelo a seguir cria um grupo de filtros que aciona uma compilação quando ela é avaliada como verdadeira. O grupo de filtros a seguir especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions.

```

import { aws_codebuild as codebuild } from 'aws-cdk-lib';
import { EventAction, FilterGroup } from "aws-cdk-lib/aws-codebuild";

const source = codebuild.Source.gitHub({
  owner: 'owner',
  repo: 'repo',
  webhook: true,
  webhookFilters: [FilterGroup.inEventOf(EventAction.WORKFLOW_JOB_QUEUED)],
});

```

Filtrar eventos de webhook de GitHub ações (Terraform)

O modelo do Terraform a seguir cria um grupo de filtros que aciona uma compilação quando ela é avaliada como verdadeira. O grupo de filtros a seguir especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions.

```

resource "aws_codebuild_webhook" "example" {
  project_name = aws_codebuild_project.example.name
  build_type   = "BUILD"
  filter_group {
    filter {
      type      = "EVENT"
      pattern   = "WORKFLOW_JOB_QUEUED"
    }
  }
}

```

```
}
}
```

Filtrar eventos de webhook de GitHub ações ()AWS CLI

Os AWS CLI comandos a seguir criam um projeto de execução de GitHub ações auto-hospedado com um grupo de filtros de solicitação de trabalho do fluxo de trabalho do GitHub Actions que aciona uma compilação quando é avaliada como verdadeira.

```
aws codebuild create-project \
--name <project name> \
--source "{\"type\":\"GITHUB\",\"location\":\"<repository location>\",\"buildspec\":\"\\\"\\\"\" \
--artifacts "{\"type\":\"NO_ARTIFACTS\""} \
--environment "{\"type\":\"LINUX_CONTAINER\",\"image\": \"aws/codebuild/amazonlinux-x86_64-standard:5.0\",\"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
--service-role "<service role ARN>"
```

```
aws codebuild create-webhook \
--project-name <project name> \
--filter-groups "[[{"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]]"
```

Solucionar problemas do webhook

Problema: o webhook que você configurou [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#) não está funcionando ou seu trabalho de fluxo de trabalho está pendente. GitHub

Causas possíveis:

- Seu evento de trabalhos do webhook Workflow pode não estar conseguindo acionar uma compilação. Revise os logs de Resposta para ver a resposta ou a mensagem de erro.
- Suas tarefas estão sendo atribuídas ao agente executor incorreto devido à configuração da etiqueta. Esse problema pode ocorrer quando um de seus trabalhos em uma única execução de fluxo de trabalho tem menos rótulos do que outro trabalho. Por exemplo, se você tiver dois trabalhos com os seguintes rótulos no mesmo fluxo de trabalho, execute:
 - Job 1: codebuild-myProject-\${{ github.run_id }}-\${{ github.run_attempt }}
 - Job 2:codebuild-myProject-\${{ github.run_id }}-\${{ github.run_attempt }}, instance-size:medium

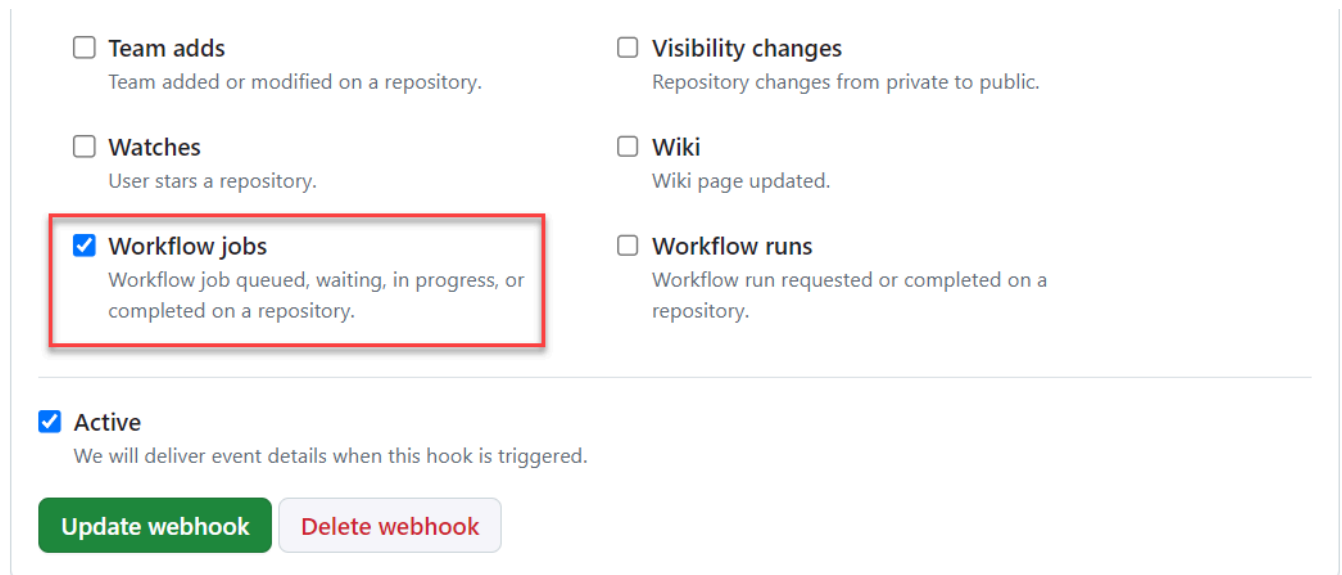
Ao rotear um trabalho de GitHub ações auto-hospedado, GitHub roteará o trabalho para qualquer executor com todos os rótulos especificados do trabalho. Esse comportamento significa que o Job 1 pode ser escolhido pelo executor criado para o Job 1 ou o Job 2, mas o Job 2 só pode ser escolhido pelo executor criado para o Job 2, pois tem um rótulo adicional. Se o Job 1 for escolhido pelo executor criado para o Job 2, o Job 2 ficará travado, pois o executor do Job 1 não tem o `instance-size:medium` rótulo.

Soluções recomendadas:

Ao criar vários trabalhos na mesma execução do fluxo de trabalho, use o mesmo número de substituições de rótulo para cada trabalho ou atribua a cada trabalho um rótulo personalizado, como `job1` ou `job2`.

Se o erro persistir, use as instruções a seguir para depurar o problema.

1. Abra o GitHub console em <https://github.com/user-name/repository-name/settings/hooks> para ver as configurações do webhook do seu repositório. Nesta página, você verá o webhook que foi criado para o repositório.
2. Escolha Edit e confirme se o webhook está habilitado para fornecer eventos de trabalhos de fluxo de trabalho.



☐ **Team adds**
Team added or modified on a repository.

☐ **Watches**
User stars a repository.

☒ **Workflow jobs**
Workflow job queued, waiting, in progress, or completed on a repository.

☐ **Visibility changes**
Repository changes from private to public.

☐ **Wiki**
Wiki page updated.

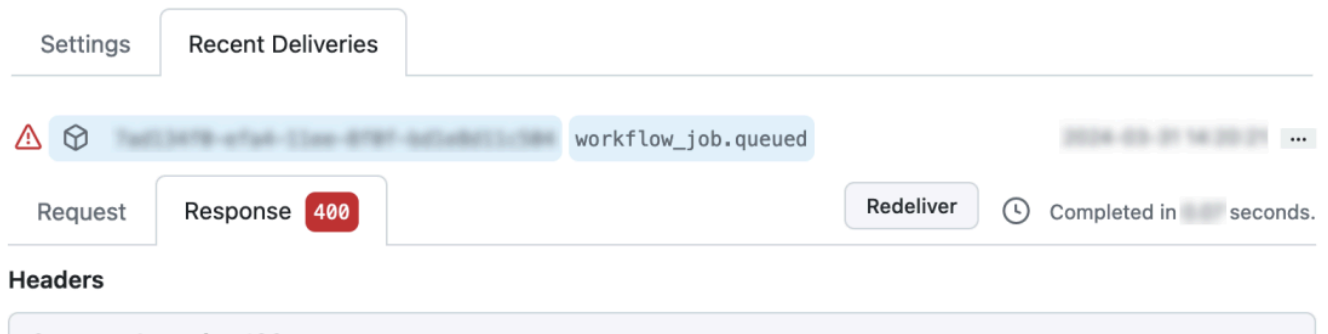
☐ **Workflow runs**
Workflow run requested or completed on a repository.

☒ **Active**
We will deliver event details when this hook is triggered.

Update webhook **Delete webhook**

3. Navegue até a guia Recent Deliveries, encontre o evento `workflow_job.queued` correspondente e expanda o evento.
4. Revise o campo de rótulos na Payload e verifique se está conforme o esperado.

5. Por fim, revise a guia Resposta, pois ela contém a resposta ou a mensagem de erro retornada de CodeBuild.



6. Como alternativa, você pode depurar falhas de webhook usando GitHub's. APIs Você pode ver as entregas recentes de um webhook usando a API [Listar entregas para um webhook de repositório](#):

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-API-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries
```

Depois de encontrar a entrega do webhook que você deseja depurar e anotar o ID de entrega, você pode usar a API [Get a delivery for a repository](#) webhook. CodeBuildA resposta da à carga útil de entrega do webhook pode ser encontrada na *response* seção:

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-API-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries/delivery-id
```

Problema: suas GitHub ações com as regras de [proteção de implantação](#) ativadas são criadas CodeBuild antes da aprovação da implantação.

Causas possíveis: CodeBuild busca a implantação e o ambiente associados à tarefa GitHub Actions, se existirem, para verificar se ela foi aprovada. Se CodeBuild não conseguir obter a implantação ou o ambiente, a CodeBuild compilação poderá ser acionada prematuramente.

Soluções recomendadas: verifique se as credenciais associadas aos seus CodeBuild projetos têm permissões de leitura para implantações e ações internas. GitHub

Substituições de rótulos suportadas pelo executor de ações CodeBuild hospedado GitHub

Em seu fluxo de trabalho de GitHub ações YAML, você pode fornecer uma variedade de substituições de rótulos que modificam sua compilação de executores auto-hospedados. Todas as compilações não reconhecidas por CodeBuild serão ignoradas, mas não falharão em sua solicitação de webhook. Por exemplo, o fluxo de trabalho YAML a seguir inclui substituições para imagem, tamanho da instância, frota e buildspec:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      image:${{ matrix.os }}
      instance-size:${{ matrix.size }}
      fleet:myFleet
      buildspec-override:true
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

Note

Se seu trabalho de fluxo de trabalho estiver suspenso GitHub, consulte [Solucionar problemas do webhook](#) e [Usando rótulos personalizados para rotear trabalhos](#).

codebuild-*<project-name>*-\${{github.run_id}}-
\${{github.run_attempt}}(obrigatório)

- Example: codebuild-fake-project-\${{ github.run_id }}-
\${{ github.run_attempt }}

- Necessário para todo o fluxo de trabalho de GitHub ações YAMLs. `<project name>` deve ser igual ao nome do projeto para o qual o webhook do executor auto-hospedado está configurado.

`image:<environment-type>-<image-identifier>`

- Example: `image:arm-3.0`
- Substitui a imagem e o tipo de ambiente usados ao iniciar a construção do executor auto-hospedado por uma imagem com curadoria. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação compatíveis com o executor CodeBuild de ações hospedado GitHub](#).
- Para substituir a imagem e o tipo de ambiente usados com uma imagem personalizada, use `image:custom-<environment-type>-<custom-image-identifier>`
- Example: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

 Note

Se a imagem personalizada residir em um registro privado, consulte [Configurar uma credencial de registro privada para executores auto-hospedados](#).

`instance-size:<instance-size>`

- Example: `instance-size:medium`
- Substitui o tipo de instância usado ao iniciar a compilação do executor auto-hospedado. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação compatíveis com o executor CodeBuild de ações hospedado GitHub](#).

`fleet:<fleet-name>`

- Example: `fleet:myFleet`
- Substitui as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#).

`buildspec-override:<boolean>`

- `Example: buildspec-override:true`
- Permite que a compilação execute comandos buildspec nas fases `INSTALL`, `PRE_BUILD` e `POST_BUILD` se definida como `true`.

Substituição de rótulo único (herdado)

CodeBuild permite que você forneça várias substituições em uma única etiqueta usando o seguinte:

- Para substituir suas configurações de ambiente para uma compilação de computação do EC2 Amazon/Lambda, use a seguinte sintaxe:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-  
${{ github.run_attempt }}-<environment-type>-<image-identifier>-<instance-size>
```

- Para substituir as configurações da sua frota para o Amazon EC2 Compute Build, use a seguinte sintaxe:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}-  
fleet-<fleet-name>
```

- Para substituir a frota e a imagem usadas para a compilação, use a seguinte sintaxe:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-  
${{ github.run_attempt }}-<image>-<image-version>-fleet-<fleet-name>
```

- Para executar comandos buildspec durante a compilação, `-with-buildspec` pode ser adicionado como sufixo ao rótulo:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-  
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>-with-buildspec
```

- Opcionalmente, você pode fornecer uma substituição do tamanho da instância sem substituir a imagem. Para EC2 compilações da Amazon, você pode excluir o tipo de ambiente e o identificador de imagem. Para compilações do Lambda, você pode excluir o identificador da imagem.

Imagens de computação compatíveis com o executor CodeBuild de ações hospedado GitHub

No rótulo em que você configurou [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#), você pode substituir as configurações do seu EC2 ambiente Amazon usando os valores nas três primeiras colunas. CodeBuild fornece as seguintes imagens de EC2 computação da Amazon. Para obter mais informações sobre

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/padrão/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/padrão/5.0
		gpu_small			
		gpu_large			
linux-ec2	latest	small	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Nenhum
		medium			
		large			
arm	2.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-armv7-standard:2.0	al/aarch64/standard/2.0
		medium			
		large			
		xlarge			

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-arm64-standard:2.0 aws/codebuild/amazonlinux-arm64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nenhum
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nenhum

Além disso, você pode substituir as configurações do ambiente Lambda usando os valores a seguir. Para obter mais informações sobre a computação CodeBuild Lambda, consulte [Execute compilações em computação AWS Lambda](#). CodeBuild é compatível com as seguintes imagens de computação Lambda:

Tipo de ambiente	Identificador da imagem	Tamanho da instância			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto1.1	4GB			
		8GB			

Tipo de ambiente	Identificador da imagem	Tamanho da instância			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Para obter mais informações, consulte [Modos e tipos de computação do ambiente de compilação e Imagens do Docker fornecidas por CodeBuild](#).

GitLab Corredores autogerenciados em AWS CodeBuild

GitLab fornece dois modos de execução para executar GitLab trabalhos em seu CI/CD pipeline. One mode is GitLab-hosted runners, which are managed by GitLab and fully integrated with GitLab. The other mode is self-managed runners, which allows you to bring your own customized environment to run jobs in the GitLab CI/CD pipeline.

As etapas de alto nível para configurar um CodeBuild projeto para executar trabalhos de pipeline de GitLab CI/CD são as seguintes:

1. Se você ainda não fez isso, conecte-se a um OAuth aplicativo ao qual conectar seu projeto GitLab.
2. Navegue até o CodeBuild console, crie um CodeBuild projeto com um webhook e configure seus filtros de webhook.
3. Atualize seu pipeline de GitLab CI/CD YAML GitLab para configurar seu ambiente de compilação.

Para obter um procedimento mais detalhado, consulte [Tutorial: Configurar um CodeBuild executor hospedado GitLab](#).

Esse recurso permite que seus trabalhos de pipeline de GitLab CI/CD obtenham integração nativa AWS, o que fornece segurança e conveniência por meio de recursos como IAM e Amazon VPC. AWS CloudTrail Você pode acessar os tipos de instância mais recentes, incluindo instâncias baseadas em ARM.

Tópicos

- [Sobre o CodeBuild -hosted runner GitLab](#)
- [Tutorial: Configurar um CodeBuild executor hospedado GitLab](#)
- [Substituições de rótulos suportadas pelo executor hospedado CodeBuild GitLab](#)
- [Imagens de computação compatíveis com o executor CodeBuild hospedado GitLab](#)

Sobre o CodeBuild -hosted runner GitLab

A seguir estão algumas perguntas comuns sobre o CodeBuild -hosted GitLab runner.

Quais tipos de fonte são compatíveis com CodeBuild GitLab executores hospedados?

CodeBuild-hosted GitLab runners são compatíveis com o tipo de GITLAB_SELF_MANAGED fonte GITLAB e.

Quando devo incluir as substituições de imagem e instância no rótulo?

Você pode incluir as substituições de imagem e instância no rótulo para especificar um ambiente de criação diferente para cada uma das suas tarefas de pipeline de GitLab CI/CD. Isso pode ser feito sem a necessidade de criar vários CodeBuild projetos ou webhooks.

Posso usar AWS CloudFormation esse recurso?

Sim, você pode incluir um grupo de filtros em seu AWS CloudFormation modelo que especifica um filtro de eventos de GitLab trabalho de fluxo de trabalho no webhook do seu projeto.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Para obter mais informações, consulte [Filtrar eventos de GitLab webhook \(\)AWS CloudFormation](#).

Se precisar de ajuda para configurar as credenciais do projeto em seu AWS CloudFormation modelo, consulte [AWS::CodeBuild::SourceCredential](#)o Guia do AWS CloudFormation usuário para obter mais informações.

Como posso mascarar segredos ao usar esse recurso?

Por padrão, os segredos impressos no log não são mascarados. Se quiser mascarar os segredos, atualize as configurações de variáveis de ambiente de CI/CD:

Para mascarar segredos em GitLab

1. Em suas GitLab Configurações, escolha CI/CD.
2. Em Variáveis, escolha Editar para o segredo que você deseja mascarar.
3. Em Visibilidade, selecione Variável de máscara e escolha Atualizar variável para salvar as alterações.

Posso receber eventos de GitLab webhook de vários projetos em um único grupo?

CodeBuild suporta webhooks de grupo, que recebem eventos de um grupo especificado GitLab . Para obter mais informações, consulte [GitLab webhooks em grupo](#).

Posso executar um trabalho no executor do Docker para o executor autogerenciado? Por exemplo, quero executar um trabalho do pipeline em uma imagem específica para manter o mesmo ambiente de compilação em um contêiner separado e isolado.

Você pode executar o executor GitLab autogerenciado CodeBuild com uma imagem específica [criando o projeto com uma imagem personalizada ou substituindo a imagem](#) em seu arquivo `.gitlab-ci.yml`

Com qual executor o executor autogerenciado trabalha? CodeBuild

O executor autogerenciado em CodeBuild é executado com o executor do shell, onde a compilação é executada localmente junto com o executor que está GitLab sendo executado dentro do contêiner docker.

Posso fornecer comandos buildspec com o executor autogerenciado?

Sim, é possível adicionar comandos buildspec com o executor autogerenciado. Você pode fornecer o arquivo buildspec.yml no seu GitLab repositório e usar a **buildspec-override:true** tag na

seção Tags do trabalho. Para obter mais informações, consulte [Nome do arquivo buildspec e local de armazenamento](#).

Quais regiões oferecem suporte ao uso de um GitLab executor CodeBuild hospedado?

CodeBuild GitLab -os executores hospedados são suportados em todas as CodeBuild regiões. Para obter mais informações sobre Regiões da AWS onde CodeBuild está disponível, consulte [AWS Serviços por região](#).

Quais plataformas oferecem suporte ao uso de um GitLab executor CodeBuild hospedado?

CodeBuild- GitLab os executores hospedados são compatíveis tanto na Amazon EC2 quanto [AWS Lambda](#) na computação. É possível usar as seguintes plataformas: Amazon Linux 2, Amazon Linux 2023, Ubuntu e Windows Server Core 2019. Para obter mais informações, consulte [EC2 imagens de computação](#) e [Imagens de computação do Lambda](#).

Tutorial: Configurar um CodeBuild executor hospedado GitLab

Este tutorial mostra como configurar seus CodeBuild projetos para executar trabalhos de pipeline de GitLab CI/CD. Para obter mais informações sobre como usar GitLab ou gerenciar automaticamente com CodeBuild, consulte [GitLab Corredores autogerenciados em AWS CodeBuild](#).

Para concluir este tutorial, você deve primeiramente:

- Conecte-se a um OAuth aplicativo usando CodeConnections. Observe que, ao se conectar a um OAuth aplicativo, você deve usar o CodeBuild console para fazer isso. Para obter mais instruções, consulte [GitLab acesso em CodeBuild](#).
- Conecte-se CodeBuild à sua GitLab conta. Para fazer isso, você pode adicionar GitLab como provedor de origem no console. Para instruções, consulte [GitLab acesso em CodeBuild](#).

Note

Isso só precisa ser feito se você não estiver conectado GitLab à sua conta. Com esse recurso, CodeBuild precisa de permissões adicionais, como `create_runner` e `manage_runner` do GitLab OAuth aplicativo. Se houver uma CodeConnections GitLab conta específica, ela não solicitará automaticamente atualizações de permissão. Para fazer isso, você pode acessar o CodeConnections console e criar uma conexão fictícia com a mesma GitLab conta para acionar a reautORIZAÇÃO e obter as permissões adicionais. Com

isso, todas as conexões existentes podem usar o recurso de execução. Depois de concluir, você poderá excluir a conexão fictícia.

Etapa 1: criar um CodeBuild projeto com um webhook

Nesta etapa, você criará um CodeBuild projeto com um webhook e o revisará no GitLab console.

Para criar um CodeBuild projeto com um webhook

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).

Em Tipo de projeto, escolha Projeto Runner.

- Em Runner:
 - Para o provedor Runner, escolha GitLab.
 - Em Credencial, selecione uma das seguintes opções:
 - Escolha Credencial de origem padrão. A conexão padrão aplica uma GitLab conexão padrão em todos os projetos.
 - Escolha Credencial de origem personalizada. A conexão personalizada aplica uma GitLab conexão personalizada que substitui as configurações padrão da sua conta.

Note

Se você ainda não criou uma conexão com seu provedor, precisará criar uma nova GitLab conexão. Para instruções, consulte [Conecte-se CodeBuild a GitLab](#).

- Em Localização do executor, escolha Repositório.
- Em Repositório, escolha o nome do seu projeto em GitLab especificando o caminho do projeto com o namespace.
- Em Ambiente:
 - Escolha uma imagem de ambiente e computação compatíveis. Observe que você tem a opção de substituir as configurações de imagem e instância usando um rótulo no YAML do pipeline de GitLab CI/CD. Para obter mais informações, consulte [Etapa 2: criar um arquivo .gitlab-ci.yml no repositório](#).

- Em Buildspec:
 - Observe que o buildspec será ignorado, a menos que `buildspec-override:true` seja adicionado como rótulo. Em vez disso, o CodeBuild substituirá para usar comandos que configurarão o executor autogerenciado.
 -
3. Continue com os valores padrão e escolha Criar projeto de compilação.
 4. Abra o GitLab console em <https://gitlab.com/user-name/repository-name/-/hooks> para verificar se um webhook foi criado e está habilitado para fornecer eventos de trabalhos do Workflow.

Etapa 2: criar um arquivo .gitlab-ci.yml no repositório

Nesta etapa, você criará um `.gitlab-ci.yml` arquivo [GitLab](#) para configurar seu ambiente de compilação e usar executores GitLab autogerenciados em. CodeBuild Para obter mais informações, consulte [Usar executores autogerenciados](#).

Atualize seu pipeline de GitLab CI/CD YAML

Navegue até <https://gitlab.com/user-name/project-name/-/tree/branch-name> e crie um arquivo `.gitlab-ci.yml` no repositório. Para configurar o ambiente de compilação, execute uma destas ações:

- Você pode especificar o nome do CodeBuild projeto. Nesse caso, a compilação usará sua configuração de projeto existente para computação, imagem, versão da imagem e tamanho da instância. O nome do projeto é necessário para vincular as configurações AWS relacionadas do seu GitLab trabalho a um CodeBuild projeto específico. Ao incluir o nome do projeto no YAML, CodeBuild é permitido invocar trabalhos com as configurações corretas do projeto.

```
tags:  
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

`$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME` é necessário para mapear a compilação para execuções de trabalhos específicos do pipeline e interromper a compilação quando a execução do pipeline é cancelada.

 Note

Certifique-se de que seu nome *<project-name>* corresponda ao nome do projeto que você criou CodeBuild. Se não corresponder, não CodeBuild processará o webhook e o pipeline de GitLab CI/CD poderá travar.

Veja a seguir um exemplo de um pipeline de GitLab CI/CD YAML:

```
workflow:
  name: HelloWorld
stages:          # List of stages for jobs, and their order of execution
  - build

build-job:       # This job runs in the build stage, which runs first.
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

- Você também pode substituir a imagem e o tipo de computação na tag. Consulte [Imagens de computação compatíveis com o executor CodeBuild hospedado GitLab](#) para obter uma lista de imagens selecionadas. Para usar imagens personalizadas, consulte [Substituições de rótulos suportadas pelo executor hospedado CodeBuild GitLab](#). O tipo de computação e a imagem na tag substituirão as configurações do ambiente em seu projeto. Para substituir suas configurações de ambiente para uma compilação EC2 computacional da Amazon, use a seguinte sintaxe:

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - image:<environment-type>-<image-identifier>
  - instance-size:<instance-size>
```

Veja a seguir um exemplo de um pipeline de GitLab CI/CD YAML:

```
stages:
  - build

build-job:
```

```
stage: build
script:
  - echo "Hello World!"
tags:
  - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - image:arm-3.0
  - instance-size:small
```

- Você pode substituir a frota usada para a compilação na tag. Isso substituirá as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#). Para substituir as configurações da sua frota para uma compilação EC2 computacional da Amazon, use a seguinte sintaxe:

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>
```

Para substituir a frota e a imagem usadas para a compilação, use a seguinte sintaxe:

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>
  - image:<environment-type>-<image-identifier>
```

Veja a seguir um exemplo de um pipeline de GitLab CI/CD YAML:

```
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:myFleet
    - image:arm-3.0
```

- Para executar seus trabalhos de pipeline de GitLab CI/CD em uma imagem personalizada, você pode configurar uma imagem personalizada em seu CodeBuild projeto e evitar fornecer um rótulo

de substituição de imagem. CodeBuild usará a imagem configurada no projeto se nenhum rótulo de substituição de imagem for fornecido.

Depois de confirmar suas alterações `.gitlab-ci.yml`, um GitLab pipeline será acionado e `build-job` ele enviará uma notificação de webhook que iniciará sua criação. CodeBuild

Execute os comandos `buildspec` nas fases `INSTALL`, `PRE_BUILD` e `POST_BUILD`

Por padrão, CodeBuild ignora qualquer comando `buildspec` ao executar uma compilação autogerenciada. GitLab Para executar comandos `buildspec` durante a compilação, `buildspec-override:true` pode ser adicionado como sufixo a tags:

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - buildspec-override:true
```

Ao usar esse comando, CodeBuild criará uma pasta chamada `gitlab-runner` na pasta de origem primária do contêiner. Quando o GitLab corredor iniciar durante a `BUILD` fase, ele será executado no `gitlab-runner` diretório.

Há várias limitações ao usar uma substituição de `buildspec` em uma compilação autogerenciada: GitLab

- CodeBuild não executará comandos `buildspec` durante a `BUILD` fase, pois o executor autogerenciado é executado na fase. `BUILD`
- CodeBuild não baixará nenhuma fonte primária ou secundária durante a `DOWNLOAD_SOURCE` fase. Se você tiver um arquivo `buildspec` configurado, somente esse arquivo será baixado da fonte primária do projeto.
- Se um comando de compilação falhar na `INSTALL` fase `PRE_BUILD` ou, o executor autogerenciado não CodeBuild iniciará e o trabalho do pipeline de GitLab CI/CD precisará ser cancelado manualmente.
- CodeBuild busca o token do executor durante a `DOWNLOAD_SOURCE` fase, que tem um tempo de expiração de uma hora. Se suas `INSTALL` fases `PRE_BUILD` ou fases excederem uma hora, o token de corredor poderá expirar antes que o corredor GitLab autogerenciado comece.

Etapa 3: revisar os resultados

Sempre que um trabalho de GitLab CI/CD pipeline run occurs, CodeBuild would receive the CI/CD pipeline job events through the webhook. For each job in the CI/CD pipeline, CodeBuild starts a build to run an ephemeral GitLab runner. The runner is responsible for executing a single CI/CD tubulação. Depois que o trabalho for concluído, o executor e o processo de compilação associado serão encerrados imediatamente.

Para visualizar os registros de trabalhos do pipeline de CI/CD, navegue até seu repositório em GitLab, escolha Build, Jobs e, em seguida, escolha o Job específico do qual você gostaria de revisar os registros.

Você pode revisar os rótulos solicitados no registro enquanto o trabalho está esperando para ser realizado por um executor autogerenciado. CodeBuild

Filtrar eventos de GitLab webhook ()AWS CloudFormation

A seguinte parte formatada em YAML de um modelo do AWS CloudFormation cria um grupo de filtros que aciona uma compilação quando ela é avaliada como verdadeira. O grupo de filtros a seguir especifica um nome de GitLab CI/CD pipeline job request with a CI/CD pipeline que corresponde à expressão `\[CI-CodeBuild\]` regular.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
    FilterGroups:
```

- - Type: EVENT
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]

Substituições de rótulos suportadas pelo executor hospedado CodeBuild GitLab

Em seu pipeline de GitLab CI/CD YAML, você pode fornecer uma variedade de substituições de rótulos que modificam sua compilação de executores autogerenciados. Todas as compilações não reconhecidas por CodeBuild serão ignoradas, mas não falharão em sua solicitação de webhook. Por exemplo, o YAML a seguir inclui substituições para imagem, tamanho da instância, frota e buildspec:

```
workflow:  
  name: HelloWorld  
stages:  
  - build  
  
build-job:  
  stage: build  
  script:  
    - echo "Hello World!"  
  tags:  
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME  
    - image:arm-3.0  
    - instance-size:small  
    - fleet:myFleet  
    - buildspec-override:true
```

codebuild-*<project-name>*-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME(obrigatório)

- Example: codebuild-myProject-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME
- Necessário para todo o GitLab pipeline de CI/CD. YAMLS *<project name>* deve ser igual ao nome do projeto para o qual o webhook do executor autogerenciado está configurado.

image:*<environment-type>*-*<image-identifier>*

- Example: image:arm-3.0

- Substitui a imagem e o tipo de ambiente usados quando a compilação do executor autogerenciado é iniciada. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação compatíveis com o executor CodeBuild hospedado GitLab](#).
- Para substituir a imagem e o tipo de ambiente usados com uma imagem personalizada, use `image:custom-<environment-type>-<custom-image-identifier>`
- Example: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

 Note

Se a imagem personalizada residir em um registro privado, consulte [Configurar uma credencial de registro privada para executores auto-hospedados](#).

`instance-size:<instance-size>`

- Example: `instance-size:small`
- Substitui o tipo de instância usado quando a compilação do executor autogerenciado é iniciada. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação compatíveis com o executor CodeBuild hospedado GitLab](#).

`fleet:<fleet-name>`

- Example: `fleet:myFleet`
- Substitui as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar compilações em frotas de capacidade reservada](#).

`buildspec-override:<boolean>`

- Example: `buildspec-override:true`
- Permite que a compilação execute comandos buildspec nas fases INSTALL, PRE_BUILD e POST_BUILD se definida como true.

Imagens de computação compatíveis com o executor CodeBuild hospedado GitLab

No rótulo em que você configurou [Tutorial: Configurar um CodeBuild executor hospedado GitLab](#), você pode substituir as configurações do seu EC2 ambiente Amazon usando os valores nas três primeiras colunas. CodeBuild fornece as seguintes imagens de EC2 computação da Amazon. Para obter mais informações sobre

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem	Definição
linux	4.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/padrão/4.0
linux	5.0	2xlarge gpu_small gpu_large	Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/padrão/5.0
linux-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Nenhum
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem	Definição
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-arm64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nenhum
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	2xlarge gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem	Definição
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nenhum

Além disso, você pode substituir as configurações do ambiente Lambda usando os valores a seguir. Para obter mais informações sobre a computação CodeBuild Lambda, consulte [Execute compilações em computação AWS Lambda](#). CodeBuild é compatível com as seguintes imagens de computação Lambda:

Tipo de ambiente	Versão Runtime	Tamanho da instância			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto1.1	4GB			
		8GB			

Tipo de ambiente	Versão Runtime	Tamanho da instância			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Para obter mais informações, consulte [Modos e tipos de computação do ambiente de compilação e Imagens do Docker fornecidas por CodeBuild](#).

Executador autogerenciado do Buildkite em AWS CodeBuild

Você pode configurar seu projeto para configurar executores do Buildkite auto-hospedados em CodeBuild contêineres para processar seus trabalhos do Buildkite. Isso pode ser feito configurando um webhook usando seu CodeBuild projeto e atualizando as etapas YAML do pipeline Buildkite para usar executores auto-hospedados hospedados em máquinas. CodeBuild

As etapas de alto nível para configurar um CodeBuild projeto para executar trabalhos do Buildkite são as seguintes:

- Navegue até o CodeBuild console e crie um CodeBuild projeto com a configuração do tipo de executor do projeto Buildkite runner
- Adicione um `job.scheduled` webhook à sua organização Buildkite.
- Atualize as etapas YAML do pipeline do Buildkite no Buildkite para configurar seu ambiente de compilação.

Para obter um procedimento mais detalhado, consulte [Tutorial: Configurar um executor CodeBuild Buildkite hospedado](#). Esse recurso permite que seus trabalhos do Buildkite obtenham integração nativa AWS, o que fornece segurança e conveniência por meio de recursos como IAM AWS CloudTrail e Amazon VPC. AWS Secrets Manager Você pode acessar os tipos de instância mais recentes, incluindo instâncias baseadas em ARM.

Sobre o executor CodeBuild Buildkite hospedado

A seguir estão algumas perguntas comuns sobre o executor CodeBuild Buildkite hospedado.

Quando devo incluir as substituições de imagem e instância no rótulo?

Você pode incluir as substituições de imagem e instância no rótulo para especificar um ambiente de compilação diferente para cada uma das suas tarefas do Buildkite. Isso pode ser feito sem a necessidade de criar vários CodeBuild projetos ou webhooks. Por exemplo, isso é útil quando você precisa usar uma [matriz para trabalhos do Buildkite](#).

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "${matrix.os}"
      instance-size: "${matrix.size}"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "a12-5.0"
        size:
          - "small"
          - "large"
```

Posso CodeBuild criar webhooks no Buildkite automaticamente?

Atualmente, o Buildkite exige que todos os webhooks sejam criados manualmente usando seu console. Você pode seguir o tutorial em [Tutorial: Configurar um executor CodeBuild Buildkite hospedado](#) para criar um webhook do Buildkite manualmente no console do Buildkite.

Posso usar AWS CloudFormation para criar webhooks do Buildkite?

AWS CloudFormation atualmente não é compatível com webhooks de execução do Buildkite, pois o Buildkite exige que os webhooks sejam criados manualmente usando seu console.

Quais regiões oferecem suporte ao uso de um executor CodeBuild Buildkite hospedado?

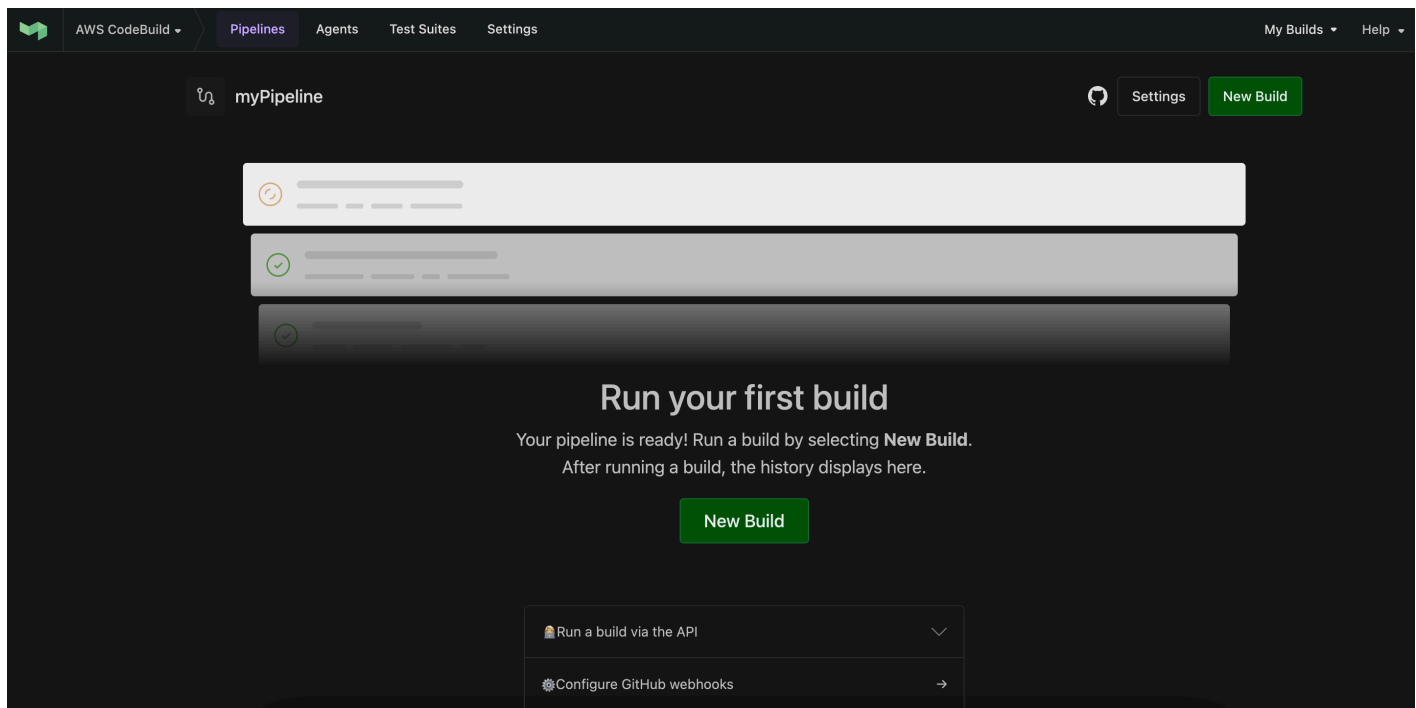
CodeBuild corredores Buildkite hospedados são suportados em todas as regiões. CodeBuild Para obter mais informações sobre AWS regiões onde CodeBuild está disponível, consulte [AWS Serviços por região](#).

Tutorial: Configurar um executor CodeBuild Buildkite hospedado

Este tutorial mostra como configurar seus CodeBuild projetos para executar trabalhos do Buildkite. Para obter mais informações sobre como usar o Buildkite com, consulte. CodeBuild [Executador autogerenciado do Buildkite em AWS CodeBuild](#)

Para concluir este tutorial, você deve primeiramente:

- Tenha acesso a uma organização Buildkite. Para obter mais informações sobre como configurar uma conta e uma organização do Buildkite, você pode seguir este tutorial de [introdução](#).
- Crie um pipeline, cluster e fila do Buildkite configurados para usar executores auto-hospedados. Para obter mais informações sobre como configurar esses recursos, consulte o tutorial de configuração do [pipeline do Buildkite](#).

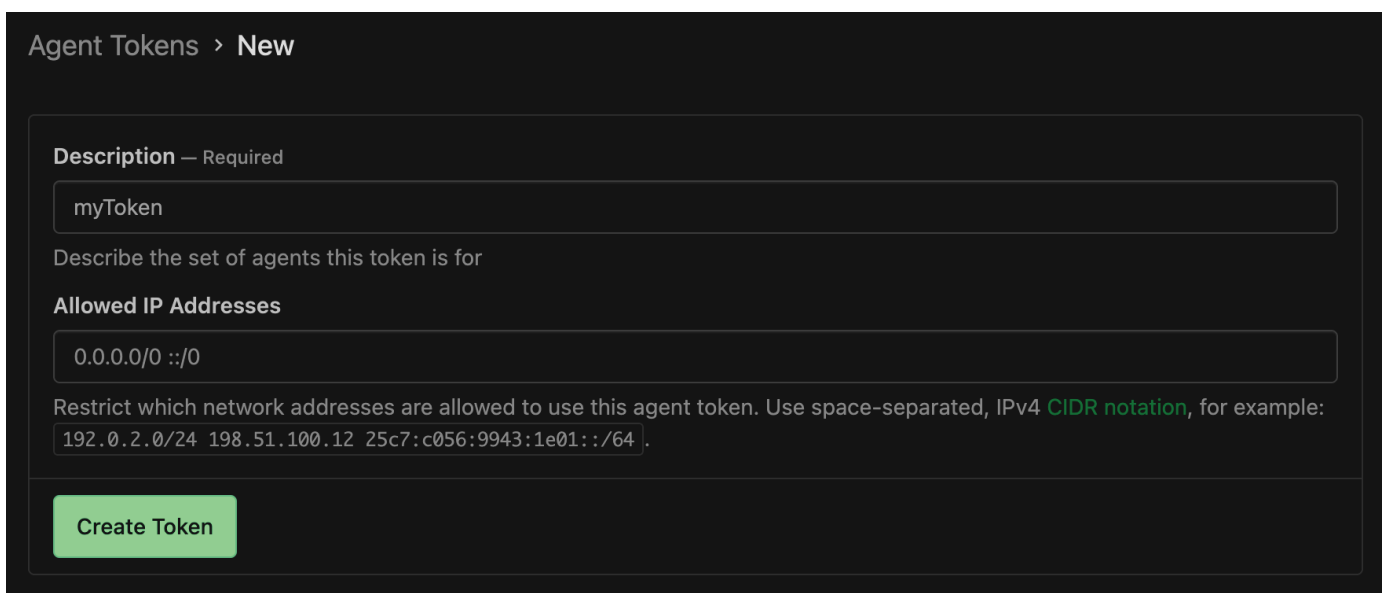


Etapa 1: gerar um token de agente Buildkite

Nesta etapa, você gerará um token de agente no Buildkite que será usado para autenticar os executores auto-hospedados. CodeBuild Para obter mais informações sobre esse recurso, consulte [Buildkite Agent Tokens](#).

Para gerar um token de agente Buildkite

1. Em seu cluster Buildkite, escolha Agent Tokens e, em seguida, escolha New Token.
2. Adicione uma descrição ao token e clique em Criar token.
3. Salve o valor do token do agente, pois ele será usado posteriormente durante a configuração do CodeBuild projeto.



The screenshot shows the 'Agent Tokens > New' form in the Buildkite interface. It has a dark theme. The form contains the following elements:

- Description — Required:** A text input field with the value 'myToken'.
- Allowed IP Addresses:** A text input field with the value '0.0.0.0/0 ::/0'.
- Instructions:** Below the IP address field, it says: 'Restrict which network addresses are allowed to use this agent token. Use space-separated, IPv4 CIDR notation, for example: 192.0.2.0/24 198.51.100.12 25c7:c056:9943:1e01::/64 .'.
- Create Token:** A green button at the bottom left of the form.

Etapa 2: criar um CodeBuild projeto com um webhook

Para criar um CodeBuild projeto com um webhook

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação auto-hospedado. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
 - Em Configuração do projeto, selecione Projeto Runner. Em Runner:
 - Para o provedor Runner, escolha Buildkite.

- Para o token do agente Buildkite, escolha Criar um novo token de agente usando a página de criação secreta. Você será solicitado a criar um novo segredo AWS Secrets Manager com um valor secreto igual ao token do agente Buildkite que você gerou acima.
- (Opcional) Se você quiser usar credenciais CodeBuild gerenciadas para seu trabalho, selecione o provedor do repositório de origem do seu trabalho nas opções de credenciais de origem do Buildkite e verifique se as credenciais estão configuradas para sua conta. Além disso, verifique se o pipeline do Buildkite usa o Checkout usando HTTPS.

 Note

O Buildkite requer credenciais de origem no ambiente de compilação para extrair a fonte do seu trabalho. Consulte as [Autenticando o Buildkite em um repositório privado](#) opções de credenciais de origem disponíveis.

- (Opcional) No ambiente:
 - Escolha uma imagem de ambiente e computação compatíveis.

Observe que você tem a opção de substituir as configurações de imagem e instância usando um rótulo nas etapas do YAML do Buildkite. Para obter mais informações, consulte [Etapa 4: atualize as etapas do pipeline do Buildkite](#).

- (Opcional) Em Buildspec:
 - Seu buildspec será ignorado por padrão, a menos que `buildspec-override: "true"` seja adicionado como um rótulo. Em vez disso, o CodeBuild substituirá para usar comandos que configurarão o executor auto-hospedado.

 Note

CodeBuild não oferece suporte a arquivos buildspec para compilações de executores auto-hospedados do Buildkite. Para especificações de construção em linha, você precisará habilitar [git-credential-helper](#) em seu buildspec se tiver configurado credenciais de origem gerenciada CodeBuild

3. Continue com os valores padrão e escolha Criar projeto de compilação.
4. Salve o URL da carga útil e os valores secretos no pop-up Create Webhook. Siga as instruções no pop-up para criar um novo webhook da organização Buildkite ou continue na próxima seção.

Etapa 3: Crie um CodeBuild webhook no Buildkite

Nesta etapa, você usará os valores Payload URL e Secret do CodeBuild webhook para criar um novo webhook no Buildkite. Esse webhook será usado para acionar compilações CodeBuild quando um trabalho válido do Buildkite for iniciado.

Para criar um novo webhook no Buildkite

1. Navegue até a página de configurações da sua organização Buildkite.
2. Em Integrações, selecione Serviços de notificação.
3. Escolha Adicionar ao lado da caixa Webhook. Na página Adicionar notificação de webhook, use a seguinte configuração:
 - a. Em URL do Webhook, adicione o valor do URL de carga útil salvo.
 - b. Em Token, verifique se a opção Enviar o token como X-Buildkite-Token está selecionada. Adicione o valor secreto do seu webhook ao campo Token.
 - c. Em, verifique se a opção Enviar o token como X-Buildkite-Token está selecionada. Adicione o valor secreto do seu webhook ao campo Token.
 - d. Em Eventos, selecione o evento do `job.scheduled` webhook.
 - e. (Opcional) Em Pipelines, você pode optar por acionar somente compilações para um pipeline específico.
4. Escolha Adicionar notificação de webhook.

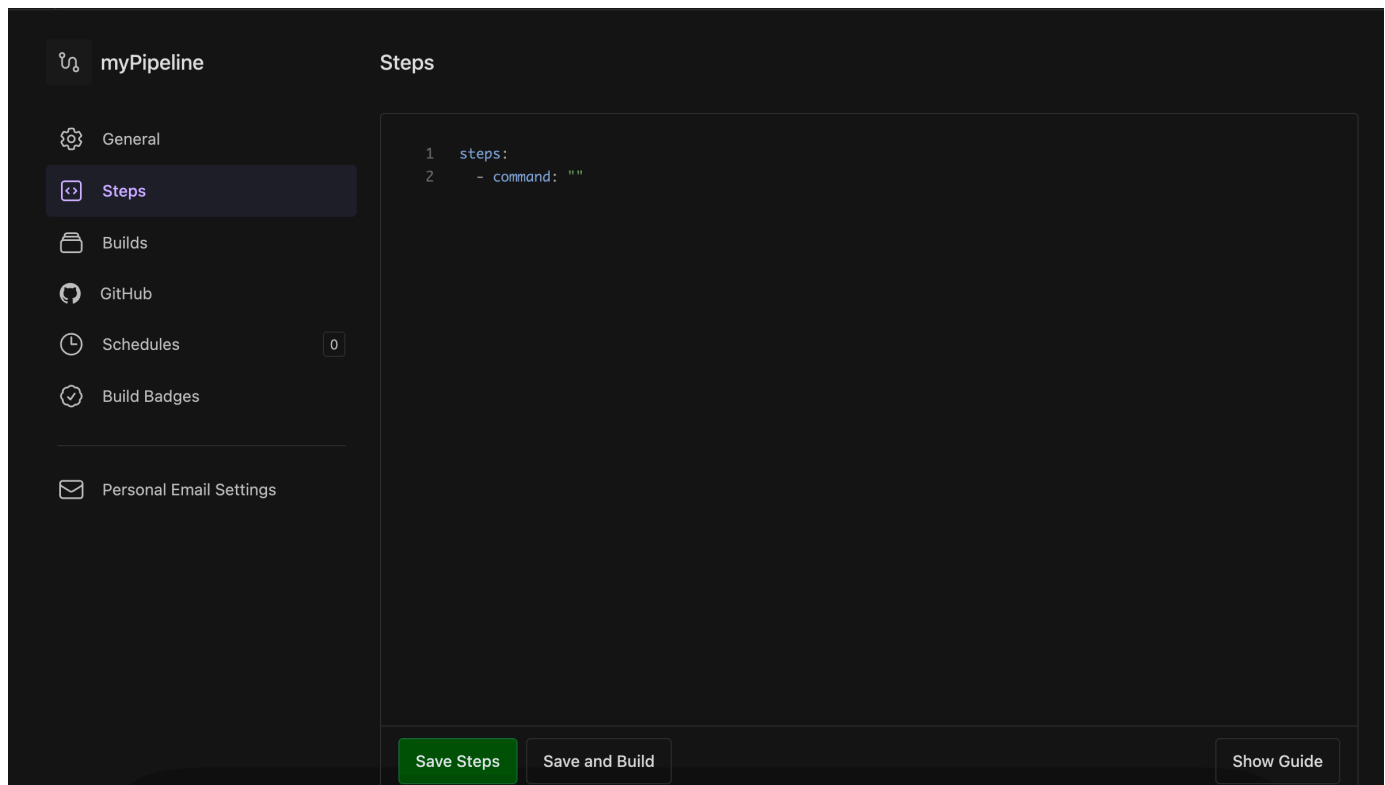
Etapa 4: atualize as etapas do pipeline do Buildkite

Nesta etapa, você atualizará as etapas do pipeline do Buildkite para adicionar os rótulos necessários e as substituições opcionais. Para ver a lista completa de substituições de rótulos compatíveis, consulte [Substituições de rótulos suportadas pelo executor Buildkite CodeBuild hospedado](#)

Atualize as etapas do seu funil

1. Navegue até a página de etapas do pipeline do Buildkite selecionando seu pipeline do Buildkite, escolhendo Configurações e, em seguida, escolhendo Etapas.

Se você ainda não tiver feito isso, escolha Converter em etapas YAML.



2. No mínimo, você precisará especificar uma [tag do agente Buildkite](#) referenciando o nome do seu pipeline. CodeBuild O nome do projeto é necessário para vincular as configurações AWS relacionadas do seu trabalho no Buildkite a um projeto específico. CodeBuild Ao incluir o nome do projeto no YAML, CodeBuild é permitido invocar trabalhos com as configurações corretas do projeto.

```
agents:
  project: "codebuild-<project name>"
```

Veja a seguir um exemplo das etapas do pipeline do Buildkite com apenas a tag do rótulo do projeto:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "echo \"Hello World\""
```

Você também pode substituir a imagem e o tipo de computação no rótulo. Para obter uma lista das imagens disponíveis, consulte [Imagens de computação suportadas pelo executor CodeBuild Buildkite hospedado](#). O tipo de computação e a imagem no rótulo substituirão as configurações

do ambiente no projeto. Para substituir suas configurações de ambiente para uma compilação de computação CodeBuild EC2 ou Lambda, use a seguinte sintaxe:

```
agents:
  project: "codebuild-<project name>"
  image: "<environment-type>-<image-identifier>"
  instance-size: "<instance-size>"
```

Veja a seguir um exemplo de etapas do pipeline do Buildkite com substituições de tamanho de imagem e instância:

```
agents:
  project: "codebuild-myProject"
  image: "arm-3.0"
  instance-size: "small"
steps:
  - command: "echo \"Hello World\""
```

Você pode substituir a frota usada para a compilação no rótulo. Isso substituirá as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar construções em frotas de capacidade reservada](#).

Para substituir as configurações da sua frota para uma compilação EC2 computacional da Amazon, use a seguinte sintaxe:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
```

Para substituir a frota e a imagem usadas para a compilação, use a seguinte sintaxe:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
  image: "<environment-type>-<image-identifier>"
```

Veja a seguir um exemplo de etapas do pipeline do Buildkite com substituições de frota e imagem:

```
agents:
  project: "codebuild-myProject"
  fleet: "myFleet"
  image: "arm-3.0"
steps:
  - command: "echo \"Hello World\""
```

3. Você pode optar por executar comandos buildspec embutidos durante a compilação do executor Buildkite auto-hospedado (consulte para obter mais detalhes). [Execute comandos buildspec para as fases INSTALL, PRE_BUILD e POST_BUILD](#) Para especificar que a CodeBuild compilação deve executar comandos buildspec durante a compilação do executor autohospedado do Buildkite, use a seguinte sintaxe:

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

Veja a seguir um exemplo de um pipeline do Buildkite com uma substituição de buildspec:

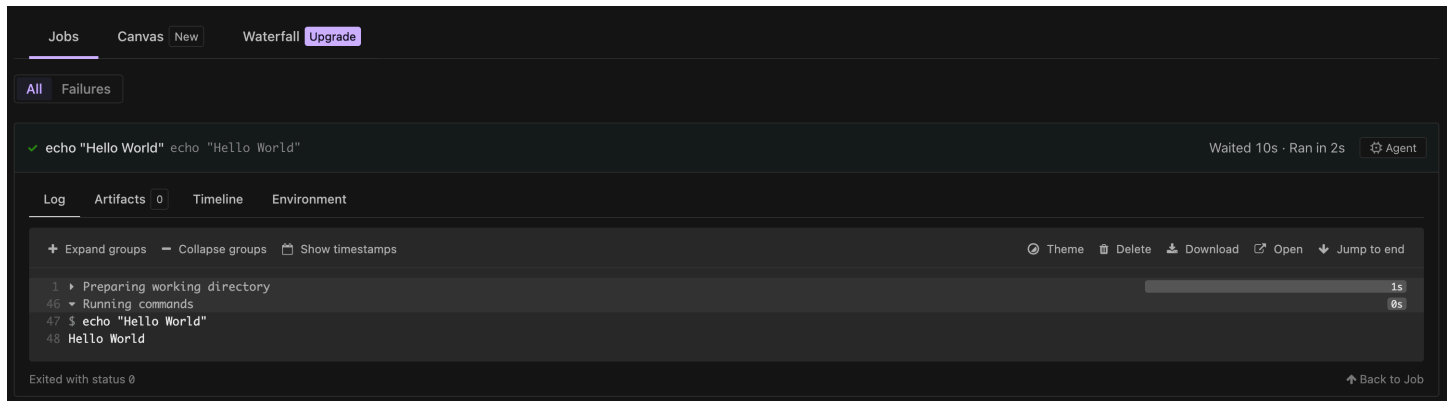
```
agents:
  project: "codebuild-myProject"
  buildspec-override: "true"
steps:
  - command: "echo \"Hello World\""
```

4. Opcionalmente, você pode fornecer rótulos fora dos CodeBuild compatíveis. Esses rótulos serão ignorados com o objetivo de substituir os atributos da compilação, mas não falharão na solicitação do webhook. Por exemplo, adicionar myLabel: "testLabel" como rótulo não impedirá a execução da compilação.

Etapa 5: analise seus resultados

Sempre que um trabalho do Buildkite for iniciado em seu pipeline, CodeBuild receberá um evento de webhook por meio do `job.scheduled` webhook do Buildkite. Para cada trabalho em sua compilação do Buildkite, CodeBuild iniciará uma compilação para executar um executor efêmero do Buildkite. O executor é responsável por executar um único trabalho do Buildkite. Depois que o trabalho for concluído, o executor e o processo de compilação associado serão encerrados imediatamente.

Para visualizar seus registros de trabalho do fluxo de trabalho, navegue até o pipeline do Buildkite e selecione a compilação mais recente (você pode acionar uma nova compilação escolhendo Nova compilação). Depois que a CodeBuild compilação associada a cada um dos seus trabalhos começar e continuar com o trabalho, você deverá ver os registros do trabalho no console do Buildkite



Autenticando o Buildkite em um repositório privado

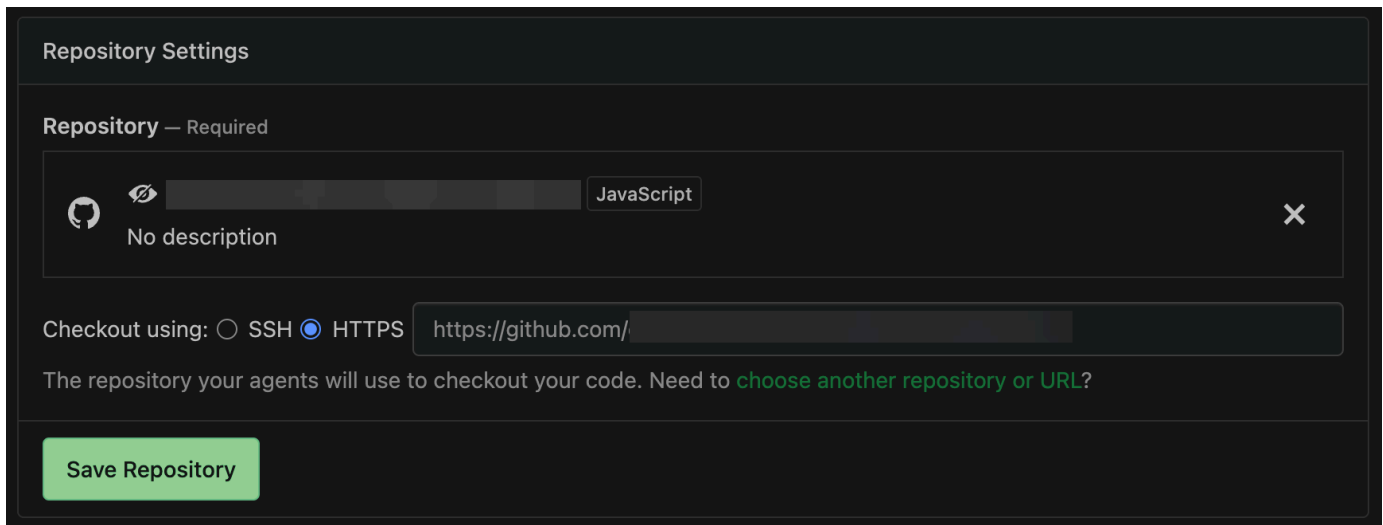
Se você tiver um repositório privado configurado em seu pipeline do Buildkite, o Buildkite exige [permissões adicionais no ambiente de compilação para extrair o repositório, pois o Buildkite](#) não vende credenciais para executores auto-hospedados extraírem de repositórios privados. Para autenticar o agente executor autohospedado do Buildkite em seu repositório externo de origem privada, você pode usar uma das seguintes opções.

Para autenticar com CodeBuild

CodeBuild oferece tratamento gerenciado de credenciais para tipos de fontes compatíveis. Para usar as credenciais de CodeBuild origem para extrair o repositório de origem do seu trabalho, você pode usar as seguintes etapas:

1. No CodeBuild console, navegue até Editar projeto ou crie um novo CodeBuild projeto usando as etapas em [Etapa 2: criar um CodeBuild projeto com um webhook](#).
2. Em Opções de credencial de origem do Buildkite, selecione o provedor do repositório de origem do seu trabalho.
 1. Se você quiser usar CodeBuild credenciais no nível da conta, verifique se elas estão configuradas corretamente. Além disso, se seu projeto tiver um buildspec embutido configurado, verifique se ele está habilitado. [git-credential-helper](#)
 2. Se você quiser usar CodeBuild credenciais em nível de projeto, selecione Usar credenciais de substituição somente para este projeto e configure credenciais para seu projeto.

3. Nas configurações do pipeline do Buildkite, navegue até Configurações do repositório. Defina as configurações de checkout do repositório de origem como Checkout usando HTTPS



Repository Settings

Repository — Required

  JavaScript 

No description

Checkout using: ☐ SSH ☒ HTTPS

The repository your agents will use to checkout your code. Need to [choose another repository or URL?](#)

Para autenticar com os segredos do Buildkite

O Buildkite mantém um [plug-in de verificação de ssh](#) que pode ser usado para autenticar o executor auto-hospedado em um repositório de origem externo usando uma chave ssh. O valor da chave é armazenado como um [segredo do Buildkite](#) e obtido automaticamente pelo agente executor auto-hospedado do Buildkite ao tentar extrair um repositório privado. Para configurar o plug-in ssh-checkout para seu pipeline Buildkite, você pode usar as seguintes etapas:

1. Gere uma chave ssh pública e privada usando seu endereço de e-mail, por exemplo `ssh-keygen -t rsa -b 4096 -C "myEmail@address.com"`
2. Adicione a chave pública ao seu repositório de origem privada. Por exemplo, você pode seguir [este guia](#) para adicionar uma chave a uma GitHub conta.
3. Adicione um [novo segredo de chave SSH](#) ao seu cluster Buildkite. Dentro do seu cluster Buildkite, selecione Segredos → Novo segredo. Adicione um nome para seu segredo no campo Chave e adicione sua chave SSH privada no campo Valor:

New Secret

Key — Required

SOURCE_SSH_KEY

Keys are case insensitive, can only contain alphanumeric and underscore characters, and can't start with BUILDKITE or BK

Value — Required

-----BEGIN OPENSSH PRIVATE KEY-----

- No pipeline do Buildkite, navegue até as configurações do repositório e defina o checkout para usar SSH.

Repository Settings

Repository — Required

  [Redacted] 
No description

Checkout using: ☒ SSH ☐ HTTPS

The repository your agents will use to checkout your code. Need to [choose another repository or URL?](#)

[Save Repository](#)

- Atualize as etapas do YAML do pipeline para usar o `git-ssh-checkout` plug-in. Por exemplo, o arquivo YAML do pipeline a seguir usa a ação de checkout com a chave secreta Buildkite acima:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "npm run build"
    plugins:
      - git-ssh-checkout#v0.4.1:
```

```
ssh-secret-key-name: 'SOURCE_SSH_KEY'
```

6. Ao executar um trabalho de execução auto-hospedado do Buildkite CodeBuild, o Buildkite agora usará automaticamente seu valor secreto configurado ao extrair seu repositório privado

Opções de configuração do executor

Você pode especificar as seguintes variáveis de ambiente na configuração do seu projeto para modificar a configuração de seus executores auto-hospedados:

- `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`: CodeBuild buscará o valor secreto configurado como o valor dessa variável de ambiente para registrar o agente AWS Secrets Manager executor auto-hospedado do Buildkite. Essa variável de ambiente deve ser do tipo `SECRETS_MANAGER` e o valor deve ser o nome do seu segredo no Secrets Manager. Uma variável de ambiente do token do agente Buildkite é necessária para todos os projetos de execução do Buildkite.
- `CODEBUILD_CONFIG_BUILDKITE_CREDENTIAL_DISABLE`: por padrão, CodeBuild carregará as credenciais de origem em nível de conta ou projeto no ambiente de compilação, pois essas credenciais são usadas pelo agente Buildkite para extrair o repositório de origem do trabalho. Para desativar esse comportamento, você pode adicionar essa variável de ambiente ao seu projeto com o valor definido como `true`, o que impedirá que as credenciais de origem sejam carregadas no ambiente de compilação.

Execute comandos `buildspec` para as fases `INSTALL`, `PRE_BUILD` e `POST_BUILD`

Por padrão, CodeBuild ignora qualquer comando `buildspec` ao executar uma compilação auto-hospedada do executor Buildkite. Para executar comandos `buildspec` durante a compilação,

```
buildspec-override: "true"
```

pode ser adicionado como um sufixo ao rótulo:

```
agents:  
  project: "codebuild-<project name>"  
  buildspec-override: "true"
```


Ao usar esse comando, CodeBuild criará uma pasta chamada `buildkite-runner` na pasta de origem primária do contêiner. Quando o executor do Buildkite iniciar durante a BUILD fase, o executor será executado no diretório `buildkite-runner`.

Há várias limitações ao usar uma substituição de buildspec em uma compilação auto-hospedada do Buildkite:

- O agente Buildkite exige que as credenciais de origem existam no ambiente de compilação para extrair o repositório de origem do trabalho. Se você usar credenciais de CodeBuild origem para autenticação, precisará habilitar `git-credential-helper` em seu buildspec. Por exemplo, você pode usar o seguinte buildspec `git-credential-helper` para habilitar suas compilações do Buildkite:

```
version: 0.2
env:
  git-credential-helper: yes
phases:
  pre_build:
    commands:
      - echo "Hello World"
```

- CodeBuild não executará comandos buildspec durante a BUILD fase, pois o executor auto-hospedado é executado na fase. BUILD
- CodeBuild não oferece suporte a arquivos buildspec para compilações de executores do Buildkite. Somente especificações de construção embutidas são suportadas para executores auto-hospedados do Buildkite
- Se um comando de compilação falhar na INSTALL fase PRE_BUILD ou, o executor auto-hospedado não CodeBuild será iniciado e a tarefa do Buildkite precisará ser cancelada manualmente.

Configurando um executor Buildkite programaticamente

Para configurar programaticamente um projeto de execução do Buildkite, você precisará configurar os seguintes recursos:

Para criar um executor Buildkite programaticamente

1. Crie um token de agente Buildkite e salve o token em texto simples dentro dele. AWS Secrets Manager

2. Configure um CodeBuild projeto com sua configuração preferida. Você precisará configurar os seguintes atributos adicionais:
 1. Um valor de ambiente com nome `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKENSECRETS_MANAGER`, tipo e um valor igual ao token do agente Buildkite associado ao seu cluster Buildkite.
 2. Tipo de fonte igual a `NO_SOURCE`
 3. Permissões para acessar o segredo criado na etapa 1 na função de serviço do seu projeto

Por exemplo, você pode usar o comando a seguir para criar um projeto de execução válido do Buildkite por meio da CLI:

```
aws codebuild create-project \
--name buildkite-runner-project \
--source "{\"type\": \"NO_SOURCE\", \"buildspec\": \"\"}" \
--environment "{\"image\": \"aws/codebuild/amazonlinux-x86_64-standard:5.0\", \"type\": \"LINUX_CONTAINER\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\", \"environmentVariables\": [{\"name\": \"CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN\", \"type\": \"SECRETS_MANAGER\", \"value\": \"<buildkite-secret-name>\"}]}" \
--artifacts "{\"type\": \"NO_ARTIFACTS\"}" \
--service-role <service-role>
```

3. Crie um webhook do Buildkite runner no projeto criado na etapa 2. Você precisará usar as seguintes opções de configuração ao criar o webhook:
 1. o tipo de construção deve ser igual a `RUNNER_BUILDKITE_BUILD`
 2. Um filtro com tipo `EVENT` e padrão iguais a `WORKFLOW_JOB_QUEUED`

Por exemplo, você pode usar o comando a seguir para criar um webhook válido do executor Buildkite por meio da CLI:

```
aws codebuild create-webhook \
--project-name buildkite-runner-project \
--filter-groups "[{\"type\": \"EVENT\", \"pattern\": \"WORKFLOW_JOB_QUEUED\"}]" \
--build-type RUNNER_BUILDKITE_BUILD
```

4. Salve o URL da carga útil e os valores secretos retornados pela `create-webhook` chamada e use as credenciais para criar um webhook no console do Buildkite. Você pode consultar a Etapa

3: Crie um CodeBuild webhook no Buildkite [Tutorial: Configurar um executor CodeBuild Buildite hospedado](#) para obter um guia sobre como configurar esse recurso.

Solucione o problema do webhook em caso de falhas de compilação ou de um trabalho suspenso

Problema:

O webhook que você configurou não [Tutorial: Configurar um executor CodeBuild Buildite hospedado](#) está funcionando ou seu trabalho de fluxo de trabalho está suspenso no Buildkite.

Causas possíveis:

- Seu evento webhook `job.scheduled` pode não estar conseguindo acionar uma compilação. Revise os logs de Resposta para ver a resposta ou a mensagem de erro.
- Sua CodeBuild compilação falha antes de iniciar o agente executor auto-hospedado do Buildkite para realizar seu trabalho.

Soluções recomendadas:

Para depurar eventos de webhook do Buildkite que falharam:

1. Nas configurações da sua organização Buildkite, navegue até os Serviços de Notificação, selecione seu CodeBuild webhook e, em seguida, encontre o Registro de Solicitações.
2. Encontre o evento de `job.scheduled` webhook associado ao seu trabalho preso no Buildkite. Você pode usar o campo ID do trabalho na carga do webhook para correlacionar o evento do webhook ao seu trabalho do Buildkite.
3. Selecione a guia Resposta e verifique o corpo da resposta. Verifique se o código de status da resposta é 200 e se o corpo da resposta não contém nenhuma mensagem inesperada.

The screenshot displays the 'Request Log' interface in AWS CodeBuild. At the top, a red box indicates a '400' status code. To its right, the event type is 'job.scheduled' and the duration is '0.29s'. The timestamp is '2024-12-11 23:03:33 UTC'. Below this, there are two tabs: 'Request' and 'Response', with 'Response' being the active tab. Under the 'Response' tab, the 'Headers' section shows the following information: 'Date: Wed, 11 Dec 2024 23:03:33 GMT', 'Connection: close', 'Content-Type: application/json', 'Content-Length: 92', 'X-Amzn-Errortype: InvalidInputException:http://', and 'X-Amzn-Requestid: 7a931bed-0bae-4c1a-9f5b-2178900cd180'. The 'Body' section shows a JSON object: {'message': 'Project name in label nonMatchingProjectName did not match actual project name'}

Solucionar os problemas de permissão do webhook

Problema:

A tarefa Buildkite falha ao verificar o repositório de origem da tarefa devido a problemas de permissão.

Causas possíveis:

- CodeBuild não tem permissões suficientes para verificar o repositório de origem do trabalho.
- As configurações do repositório do pipeline estão definidas para serem finalizadas usando SSH para credenciais CodeBuild gerenciadas.

Soluções recomendadas:

- Verifique se CodeBuild tem permissões suficientes configuradas para verificar o repositório de origem do trabalho. Além disso, verifique se a função de serviço do seu CodeBuild projeto tem permissões suficientes para acessar a opção de permissão de origem configurada.
- Verifique se seu pipeline do Buildkite está configurado para usar o checkout usando HTTPS se você estiver usando credenciais de repositório de origem CodeBuild gerenciada.

Substituições de rótulos suportadas pelo executor Buildkite CodeBuild hospedado

Nos rótulos de tag de agente das etapas do pipeline do Buildkite, você pode fornecer uma variedade de substituições de rótulos que modificam sua compilação de executores autohospedados. Todas as compilações não reconhecidas por CodeBuild serão ignoradas, mas não falharão em sua solicitação de webhook. Por exemplo, o fluxo de trabalho YAML a seguir inclui substituições para imagem, tamanho da instância, frota e buildspec:

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
      buildspec-override: "true"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "a12-5.0"
        size:
          - "small"
          - "large"
```

project:codebuild-*<project-name>*(obrigatório)

- Example: project: "codebuild-myProject"
- Necessário para todas as configurações de etapas do pipeline do Buildkite. *<project name>* deve ser igual ao nome do projeto para o qual o webhook do executor auto-hospedado está configurado.

queue: "*<queue-name>*"

- Example: queue: "*<queue-name>*"
- Usado para rotear trabalhos do Buildkite para uma fila específica. Consulte a [etiqueta Buildkite Agent Queue](#) para obter mais informações.

image: "<environment-type>-<image-identifier>"

- Example: image: "arm-3.0"
 - Substitui a imagem e o tipo de ambiente usados ao iniciar a construção do executor auto-hospedado por uma imagem com curadoria. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação suportadas pelo executor CodeBuild Buildkite hospedado](#).
1. Para substituir a imagem e o tipo de ambiente usados com uma imagem personalizada, use
image: "custom-<environment-type>-<custom-image-identifier>"
 2. Exemplo:

```
image:
  "custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0"
```

 Note

Se a imagem personalizada residir em um registro privado, você deverá configurar as credenciais de registro apropriadas em seu CodeBuild projeto.

instance-size: "<instance-size>"

- Example: instance-size: "medium"
- Substitui o tipo de instância usado ao iniciar a compilação do executor auto-hospedado. Para saber mais sobre os valores compatíveis, consulte [Imagens de computação suportadas pelo executor CodeBuild Buildkite hospedado](#).

fleet: "<fleet-name>"

- Example: fleet: "myFleet"
- Substitui as configurações de frota definidas no projeto para usar a frota especificada. Para obter mais informações, consulte [Executar construções em frotas de capacidade reservada](#).

buildspec-override: "<boolean>"

- Example: buildspec-override: "true"

- Permite que a compilação execute comandos `buildspec` nas fases `INSTALL`, `PRE_BUILD` e `POST_BUILD` se definida como `true`.

Imagens de computação suportadas pelo executor CodeBuild Buildkite hospedado

No rótulo em que você configurou [Executador autogerenciado do Buildkite em AWS CodeBuild](#), você pode substituir as configurações do seu EC2 ambiente Amazon usando os valores nas três primeiras colunas. CodeBuild fornece as seguintes imagens de EC2 computação da Amazon. Para obter mais informações sobre

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
linux	4.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/padrão/4.0
linux	5.0	2xlarge gpu_small gpu_large	Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/padrão/5.0
linux-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Nenhum
arm	2.0	small medium	Amazon Linux 2	aws/codebuild/amazonlinux-a	al/aarch64/standard/2.0

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
		large xlarge		arch64-standard:2.0	
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-arm64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nenhum
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	2xlarge gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D

Tipo de ambiente	Identificador da imagem	Tamanho da instância	Plataforma	Imagem resolvida	Definição
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nenhum

Além disso, você pode substituir as configurações do ambiente Lambda usando os valores a seguir. Para obter mais informações sobre a computação CodeBuild Lambda, consulte [Execute compilações em computação AWS Lambda](#). CodeBuild é compatível com as seguintes imagens de computação Lambda:

Tipo de ambiente	Identificador da imagem	Tamanho da instância			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto1	4GB			
	1	8GB			

Tipo de ambiente	Identificador da imagem	Tamanho da instância			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Para obter mais informações, consulte [Modos e tipos de computação do ambiente de compilação e Imagens do Docker fornecidas por CodeBuild](#).

Use webhooks com AWS CodeBuild

AWS CodeBuild suporta integração de webhook com GitHub GitHub Enterprise Server GitLab, GitLab Self Managed e Bitbucket.

Tópicos

- [Práticas recomendadas para usar webhooks com o AWS CodeBuild](#)
- [Filtrar eventos de webhook do Bitbucket](#)
- [GitHub webhooks globais e organizacionais](#)
- [GitHub webhooks manuais](#)
- [GitHub eventos de webhook](#)
- [GitLab webhooks em grupo](#)
- [GitLab webhooks manuais](#)

- [GitLab eventos de webhook](#)
- [Webhooks manuais do Buildkite](#)

Práticas recomendadas para usar webhooks com o AWS CodeBuild

Para projetos que usam repositórios públicos para configurar webhooks, recomendamos as seguintes opções:

Filtros de configuração ACTOR_ACCOUNT_ID

Adicione filtros ACTOR_ACCOUNT_ID aos grupos de filtros de webhook do projeto para especificar quais usuários podem acionar uma compilação. Cada evento de webhook entregue CodeBuild vem com informações do remetente que especificam o identificador do ator. CodeBuild filtrará os webhooks com base no padrão de expressão regular fornecido nos filtros. É possível determinar os usuários específicos que têm permissão para acionar compilações com esse filtro. Para ter mais informações, consulte [GitHub eventos de webhook](#) e [Filtrar eventos de webhook do Bitbucket](#).

Filtros de configuração FILE_PATH

Adicione filtros FILE_PATH aos grupos de filtros de webhook do projeto para incluir ou excluir os arquivos que podem acionar uma compilação quando alterados. Por exemplo, é possível negar solicitações de compilação para alterações no arquivo `buildspec.yml` usando um padrão de expressão regular, como `^buildspec.yml$`, junto com a propriedade `excludeMatchedPattern`. Para ter mais informações, consulte [GitHub eventos de webhook](#) e [Filtrar eventos de webhook do Bitbucket](#).

Definir o escopo das permissões para o perfil do IAM de compilação

As compilações acionadas por um webhook usam o perfil de serviço do IAM especificado no projeto. Recomendamos definir as permissões no perfil de serviço com o conjunto mínimo de permissões necessário para executar a compilação. Por exemplo, em um cenário de teste e implantação, crie um projeto para teste e outro projeto para implantação. O projeto de teste aceita compilações de webhook do repositório, mas não fornece permissões de gravação para os recursos. O projeto de implantação fornece permissões de gravação para os recursos, e o filtro de webhook está configurado para permitir que somente usuários confiáveis acionem compilações.

Usar um buildspec em linha ou armazenado no Amazon S3

Se você definir o buildspec em linha dentro do próprio projeto ou armazenar o arquivo buildspec em um bucket do Amazon S3, o arquivo buildspec estará visível somente para o proprietário do projeto. Isso evita que solicitações pull façam alterações no código do arquivo buildspec e acionem compilações indesejadas. Para obter mais informações, consulte [ProjectSource.buildspec](#) na Referência da API. CodeBuild

Filtrar eventos de webhook do Bitbucket

Você pode usar grupos de filtros de webhook para especificar quais eventos de webhook do Bitbucket acionam uma compilação. Por exemplo, é possível especificar que uma compilação seja acionada somente para alterações em ramificações especificadas.

Você pode criar um ou mais grupos de filtros de webhook para especificar quais eventos de webhook acionam uma compilação. Uma compilação será acionada se algum grupo de filtros for avaliado como verdadeiro, o que ocorre quando todos os filtros no grupo são avaliados como verdadeiros. Ao criar um grupo de filtros, é necessário especificar:

Um evento

Para o Bitbucket, é possível escolher um ou mais dos seguintes eventos:

- PUSH
- PULL_REQUEST_CREATED
- PULL_REQUEST_UPDATED
- PULL_REQUEST_MERGED
- PULL_REQUEST_CLOSED

O tipo de evento do webhook está em seu cabeçalho no campos X-Event-Key. A tabela a seguir mostra como valores de cabeçalho X-Event-Key são mapeados para os tipos de eventos.

Note

Você deve habilitar o evento merged em sua configuração de webhook do Bitbucket caso você crie um grupo de filtros de webhook que use o tipo de evento PULL_REQUEST_MERGED. Você também deve habilitar o evento declined em sua

configuração de webhook do Bitbucket caso você crie um grupo de filtros do webhook que usa o tipo de evento `PULL_REQUEST_CLOSED`.

Valor do cabeçalho X-Event-Key	Tipo de evento
<code>repo:push</code>	<code>PUSH</code>
<code>pullrequest:created</code>	<code>PULL_REQUEST_CREATED</code>
<code>pullrequest:updated</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pullrequest:fulfilled</code>	<code>PULL_REQUEST_MERGED</code>
<code>pullrequest:rejected</code>	<code>PULL_REQUEST_CLOSED</code>

Para `PULL_REQUEST_MERGED`, se uma solicitação pull for mesclada com a estratégia squash e a ramificação da solicitação pull for fechada, a confirmação original da solicitação pull deixará de existir. Nesse caso, a variável de ambiente `CODEBUILD_WEBHOOK_MERGE_COMMIT` contém o identificador da confirmação de mesclagem comprimida.

Um ou mais filtros opcionais

Use uma expressão regular para especificar um filtro. Para um evento acionar uma compilação, cada filtro do grupo associado a ele deve ser avaliado como verdadeiro.

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` no console)

Um evento de webhook aciona uma compilação quando um ID de conta do Bitbucket corresponde ao padrão da expressão regular. Esse valor é exibido na propriedade `account_id` do objeto `actor` na carga de filtro webhook.

`HEAD_REF`

Um evento de webhook aciona uma compilação quando a referência head corresponde ao padrão da expressão regular (por exemplo, `refs/heads/branch-name` e `refs/tags/tag-name`). Um filtro `HEAD_REF` avalia o nome de referência do Git para a ramificação ou tag. O nome de ramificação ou de tag é exibido no campo `name` do objeto `new` no objeto `push` da carga webhook. Para eventos de solicitação pull, o nome da ramificação é exibido no campo `name` no objeto `branch` do objeto `source` na carga webhook.

BASE_REF

Um evento de webhook aciona uma compilação quando a referência base corresponde ao padrão da expressão regular. Um filtro BASE_REF funciona apenas com eventos de solicitação pull (por exemplo, refs/heads/branch-name). Um filtro BASE_REF avalia o nome de referência do Git para a ramificação. O nome da ramificação é exibido no campo name do objeto branch no objeto destination na carga webhook.

FILE_PATH

Um evento de webhook aciona uma compilação quando o caminho de um arquivo alterado corresponde ao padrão da expressão regular.

COMMIT_MESSAGE

Um webhook aciona uma compilação quando a mensagem de confirmação head corresponde à expressão padrão.

WORKFLOW_NAME

Um webhook aciona uma compilação quando o nome do fluxo de trabalho corresponde ao padrão da expressão regular.

Note

Você pode encontrar a carga webhook nas configurações de webhook no seu repositório do Bitbucket.

Tópicos

- [Filtrar eventos de webhook do Bitbucket \(console\)](#)
- [Filtrar eventos de webhook do Bitbucket \(SDK\)](#)
- [Filtrar eventos de webhook do Bitbucket \(AWS CloudFormation\)](#)

Filtrar eventos de webhook do Bitbucket (console)

Para usar o AWS Management Console para filtrar eventos de webhook:

1. Selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório) ao criar seu projeto.

2. Em Event type (Tipo de evento), escolha um ou mais eventos.
3. Para filtrar quando um evento aciona uma compilação, em Start a build under these conditions (Iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
4. Para filtrar quando um evento não é acionado, em Don't start a build under these conditions (Não iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
5. Escolha Add filter group (Adicionar grupo de filtros) para adicionar outro grupo de filtros.

Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [WebhookFilter](#) na Referência da API do AWS CodeBuild .

Neste exemplo, um grupo de filtros de webhook aciona uma compilação apenas para solicitações pull:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

► Start a build under these conditions - optional

► Don't start a build under these conditions - optional

Usando um exemplo de dois grupos de filtros, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas ou atualizadas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/branch1!`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para todas as solicitações, com exceção de eventos de tag.

Filter group 1

Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

► Start a build under these conditions - optional

▼ Don't start a build under these conditions - optional

Add filter

Filter 1

Type

HEAD_REF

Pattern

^refs/tags/.*

Neste exemplo, um grupo de filtros de webhook acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular `^buildspec.*` forem alterados.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando os arquivos são alterados nas pastas `src` ou `test`.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando uma alteração é feita por um usuário do Bitbucket que não tem um ID da conta que corresponda à expressão regular `actor-account-id`.

 Note

Para obter informações sobre como encontrar o ID da sua conta do Bitbucket, consulte <https://api.bitbucket.org/2.0/users/user-name>, onde *user-name* está seu nome de usuário do Bitbucket.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

PULL_REQUEST_CLOSED ✕

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para um evento de push quando a mensagem de confirmação head corresponde à expressão regular `\[CodeBuild\]`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Filtrar eventos de webhook do Bitbucket (SDK)

Para usar o AWS CodeBuild SDK para filtrar eventos de webhook, use o `filterGroups` campo na sintaxe de solicitação dos métodos da API `CreateWebhook` ou `UpdateWebhook`. Para obter mais informações, consulte [WebhookFilter](#) na Referência de APIs do CodeBuild .

Para criar um filtro de webhook que acione uma compilação apenas para solicitações pull, insira o seguinte na sintaxe de solicitação:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    }
  ]
]
```

Para criar um filtro de webhook que acione uma compilação apenas para ramificações especificadas, use o parâmetro `pattern` para especificar uma expressão regular a fim de filtrar os nomes da ramificação. Usando um exemplo de dois grupos de filtros, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas ou atualizadas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/myBranch$`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Você pode usar o parâmetro `excludeMatchedPattern` para especificar quais eventos não acionam uma compilação. Neste exemplo, uma compilação é acionada para todas as solicitações, com exceção de eventos de tag.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```

```

    "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
  },
  {
    "type": "HEAD_REF",
    "pattern": "^refs/tags/.*",
    "excludeMatchedPattern": true
  }
]
]
```

Você pode criar um filtro que acione uma compilação somente quando uma alteração é feita por um usuário do Bitbucket com ID da conta `actor-account-id`.

Note

Para obter informações sobre como encontrar o ID da sua conta do Bitbucket, consulte <https://api.bitbucket.org/2.0/users/user-name>, onde *user-name* está seu nome de usuário do Bitbucket.

```

"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

Você pode criar um filtro que acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular no argumento `pattern` forem alterados. Neste exemplo, o grupo de filtros especifica que uma compilação será acionada apenas quando arquivos com um nome que corresponda à expressão regular `^buildspec.*` forem alterados.

```

"filterGroups": [
```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "FILE_PATH",
    "pattern": "^buildspec.*"
  }
]
```

Neste exemplo, um grupo de filtros especifica que uma compilação seja acionada somente quando os arquivos são alterados nas pastas `src` ou `test`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

É possível criar um filtro que aciona uma compilação somente quando a mensagem de confirmação `head` corresponde à expressão regular no argumento padrão. Neste exemplo, o grupo de filtros especifica que uma compilação é acionada somente quando a mensagem de confirmação `head` do evento de `push` corresponde à expressão regular `\[CodeBuild\]`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "COMMIT_MESSAGE",
      "pattern": "\[CodeBuild\]"
    }
  ]
]
```

```
    }  
  ]  
]
```

Filtrar eventos de webhook do Bitbucket (AWS CloudFormation)

Para usar um AWS CloudFormation modelo para filtrar eventos de webhook, use a `FilterGroups` propriedade do AWS CodeBuild projeto. A seguinte parte formatada em YAML de um modelo do AWS CloudFormation cria dois grupos de filtros. Juntos, eles acionarão uma compilação quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas ou atualizadas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` por um usuário do Bitbucket que não tem o ID de conta 12345.
- O segundo grupo de filtros especifica solicitações push que são criadas em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/.*`.
- O terceiro grupo de filtros especifica uma solicitação push com uma mensagem de confirmação head correspondente à expressão regular `\[CodeBuild\]`.

```
CodeBuildProject:  
  Type: AWS::CodeBuild::Project  
  Properties:  
    Name: MyProject  
    ServiceRole: service-role  
    Artifacts:  
      Type: NO_ARTIFACTS  
    Environment:  
      Type: LINUX_CONTAINER  
      ComputeType: BUILD_GENERAL1_SMALL  
      Image: aws/codebuild/standard:5.0  
    Source:  
      Type: BITBUCKET  
      Location: source-location  
    Triggers:  
      Webhook: true  
      FilterGroups:  
        - Type: EVENT  
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED  
        - Type: BASE_REF  
          Pattern: ^refs/heads/main$
```



```
ExcludeMatchedPattern: false
- Type: ACTOR_ACCOUNT_ID
  Pattern: 12345
  ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
- Type: HEAD_REF
  Pattern: ^refs/heads/.+
- Type: FILE_PATH
  Pattern: README
  ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
- Type: COMMIT_MESSAGE
  Pattern: \[CodeBuild\]
- Type: FILE_PATH
  Pattern: ^src/.+|^test/.+
```

GitHub webhooks globais e organizacionais

Você pode usar webhooks CodeBuild GitHub globais ou organizacionais para iniciar compilações em eventos de webhook de qualquer repositório dentro de uma organização ou empresa. GitHub Os webhooks globais e organizacionais funcionam com qualquer um dos tipos de eventos de GitHub webhook existentes e podem ser configurados adicionando uma configuração de escopo ao criar um webhook. CodeBuild Você também pode usar webhooks globais e organizacionais para [configurar executores de GitHub ação auto-hospedados a fim de receber WORKFLOW_JOB_QUEUED eventos de vários repositórios CodeBuild em](#) um único projeto.

Tópicos

- [Configure um GitHub webhook global ou organizacional](#)
- [Filtrar eventos de webhook GitHub globais ou organizacionais \(console\)](#)
- [Filtrar eventos de webhook da GitHub organização \(\)AWS CloudFormation](#)

Configure um GitHub webhook global ou organizacional

As etapas de alto nível para configurar um GitHub webhook global ou organizacional são as seguintes. Para obter mais informações sobre GitHub webhooks globais e organizacionais, consulte [GitHub webhooks globais e organizacionais](#)

1. Defina o local de origem do seu projeto como `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. Na configuração do escopo do webhook, defina o escopo como `GITHUB_ORGANIZATION` ou `GITHUB_GLOBAL`, dependendo se ele deve ser uma organização ou um [webhook global](#). Para obter mais informações, consulte [Tipos de webhooks](#).
3. Especifique um nome como parte da configuração do escopo do webhook. Para webhooks organizacionais, esse é o nome da organização e, para webhooks globais, esse é o nome da empresa.

 Note

Se o tipo de origem do projeto for `GITHUB_ENTERPRISE`, você também precisará especificar um domínio como parte da configuração do escopo do webhook.

4. (Opcional) Se você quiser receber apenas eventos de webhook para repositórios específicos em sua organização ou empresa, você pode especificar `REPOSITORY_NAME` como filtro ao criar o webhook.
5. Se você estiver criando um webhook organizacional, certifique-se de que ele CodeBuild tenha permissões para criar webhooks no nível da organização. GitHub Você pode criar um token de acesso GitHub pessoal com permissões de webhook da organização ou usar CodeBuild OAuth. Para obter mais informações, consulte [GitHub e token de acesso do GitHub Enterprise Server](#).

Observe que os webhooks da organização funcionam com qualquer um dos tipos de eventos de GitHub webhook existentes.

6. Se você estiver criando um webhook global, o webhook precisará ser criado manualmente. Para obter mais informações sobre como criar manualmente um webhook em GitHub, consulte [GitHub webhooks manuais](#).

Observe que os webhooks globais são compatíveis apenas com o tipo de evento `WORKFLOW_JOB_QUEUED`. Para obter mais informações, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

Filtrar eventos de webhook GitHub globais ou organizacionais (console)

Ao criar um GitHub projeto por meio do console, selecione as seguintes opções para criar um webhook GitHub global ou organizacional dentro do projeto. Para obter mais informações sobre GitHub webhooks globais e organizacionais, consulte [GitHub webhooks globais e organizacionais](#)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
 2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
- Em Origem:
 - Em Source provider, escolha GitHubGitHubEnterprise.
 - Em Repositório, escolha webhook com GitHubescopo definido.

O GitHub repositório será automaticamente definido comoCODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION, que é o local de origem necessário para webhooks globais e organizacionais.

 Note

Se você estiver usando webhooks da organização, verifique se ele CodeBuild tem permissões para criar webhooks no nível da organização. GitHub Se você estiver usando uma [OAuthconexão existente](#), talvez seja necessário regenerar a conexão para conceder CodeBuild essa permissão. Como alternativa, você pode criar o webhook manualmente usando o recurso de [webhooks CodeBuild manuais](#). Observe que, se você tiver um GitHub OAuth token existente e quiser adicionar permissões adicionais à organização, poderá [revogar a permissão do OAuth token](#) e reconectá-lo por meio do CodeBuild console.

Source**Add source****Source 1 - Primary**

Source provider

GitHub ▼

Repository

☐ Repository in my GitHub account☐ Public repository☒ GitHub scoped webhook

GitHub repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

Connection status

You are connected to GitHub using a personal access token.

Disconnect from GitHub

- Em Eventos de webhook da origem principal:
 - Em Tipo de escopo, escolha Nível da organização se você estiver criando um webhook organizacional ou Nível corporativo se estiver criando um webhook global.
 - Em Nome, insira o nome da empresa ou da organização, dependendo se o webhook é global ou organizacional.

Se o tipo de origem do projeto for `GITHUB_ENTERPRISE`, você também precisará especificar um domínio como parte da configuração da organização do webhook. Por exemplo, se o URL da organização for **`https://domain.com/orgs/org-name`**, o domínio será **`https://domain.com`**.

Note

Não é possível alterar o nome depois que o webhook é criado. Para alterar o nome, você pode excluir e recriar o webhook. Se quiser remover completamente o webhook, você também pode atualizar o local de origem do projeto em um GitHub repositório.

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#) ☒ Rebuild every time a code change is pushed to this repository

Scope type

☒ Organization level☐ Enterprise level

Organization name

Your GitHub organization name.

Build type

☒ Single build
Triggers single build☐ Batch build
Triggers multiple builds as single execution**► Additional configuration**

- (Opcional) Nos grupos de filtros de eventos de Webhook, você pode especificar quais [eventos você gostaria de acionar uma nova compilação](#). Você também pode especificar `REPOSITORY_NAME` como filtro para acionar somente compilações em eventos de webhook de repositórios específicos.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Você também pode definir o tipo de evento WORKFLOW_JOB_QUEUED para configurar executores de GitHub ações auto-hospedados. Para obter mais informações, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

- Continue com os valores padrão e escolha Criar projeto de compilação.

Filtrar eventos de webhook da GitHub organização ()AWS CloudFormation

Para usar um AWS CloudFormation modelo para filtrar eventos de webhook da organização, use a ScopeConfiguration propriedade do AWS CodeBuild projeto. Para obter mais informações sobre GitHub webhooks globais e organizacionais, consulte. [GitHub webhooks globais e organizacionais](#)

Note

Webhooks globais e webhooks GitHub corporativos não são suportados pelo. AWS CloudFormation

A seguinte parte formatada em YAML de um AWS CloudFormation modelo cria quatro grupos de filtros. Juntos, eles acionarão uma compilação quando um ou todos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica que as pull requests são criadas ou atualizadas em ramificações com nomes de referência do Git que correspondem à expressão regular `^refs/heads/main$` expressão regular de GitHub um usuário que não tem ID de conta. 12345
- O segundo grupo de filtros especifica que as solicitações push sejam criadas em arquivos com nomes que correspondem à expressão regular `READ_ME`, em ramificações com nomes de referência do Git que correspondem à expressão regular `^refs/heads/.*`.
- O terceiro grupo de filtros especifica uma solicitação push com uma mensagem de confirmação head correspondente à expressão regular `\[CodeBuild\]`.
- O quarto grupo de filtros especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions com um nome de fluxo de trabalho correspondente à expressão `\[CI-CodeBuild\]` regular.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
```

```
    Pattern: ^refs/heads/main$
    ExcludeMatchedPattern: false
-   Type: ACTOR_ACCOUNT_ID
    Pattern: 12345
    ExcludeMatchedPattern: true
-   Type: EVENT
    Pattern: PUSH
-   Type: HEAD_REF
    Pattern: ^refs/heads/.+
-   Type: FILE_PATH
    Pattern: READ_ME
    ExcludeMatchedPattern: true
-   Type: EVENT
    Pattern: PUSH
-   Type: COMMIT_MESSAGE
    Pattern: \[CodeBuild\]
-   Type: FILE_PATH
    Pattern: ^src/.+|^test/.+
-   Type: EVENT
    Pattern: WORKFLOW_JOB_QUEUED
-   Type: WORKFLOW_NAME
    Pattern: \[CI-CodeBuild\]
```

GitHub webhooks manuais

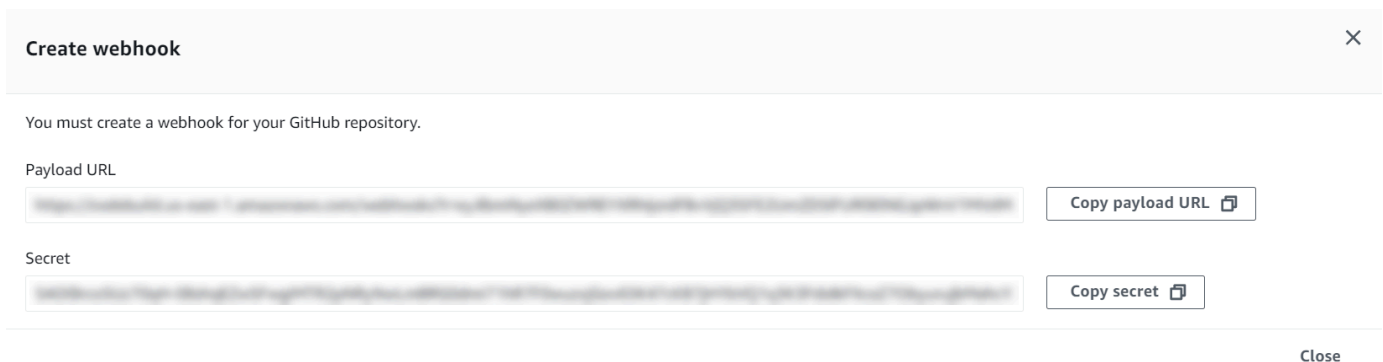
Você pode configurar GitHub webhooks manuais para evitar a tentativa automática CodeBuild de criar um webhook nele. GitHub CodeBuild retorna um URL de carga útil como parte da chamada para criar o webhook e pode ser usado para criar manualmente o webhook dentro dele. Mesmo que não CodeBuild esteja na lista de permissões para criar um webhook em sua GitHub conta, você ainda pode criar manualmente um webhook para seu projeto de compilação.

Use o procedimento a seguir para criar um webhook GitHub manual.

Para criar um GitHub webhook manual

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
 - Em Origem:
 - Em Source provider, escolha GitHub.

- Em Repositório, escolha Repositório em minha GitHub conta.
 - Em URL do repositório, insira **`https://github.com/user-name/repository-name`**.
 - Em Eventos de webhook da origem principal:
 - Em Webhook - opcional, selecione Recompilar toda vez que uma alteração de código for enviada para este repositório.
 - Escolha Configuração adicional e, para Criação manual - opcional, escolha Criar manualmente um webhook para este repositório no GitHub console. .
3. Continue com os valores padrão e escolha Criar projeto de compilação. Anote o URL da carga útil e os valores do Segredo, pois serão usados posteriormente.



Create webhook

You must create a webhook for your GitHub repository.

Payload URL

Copy payload URL

Secret

Copy secret

Close

4. Abra o GitHub console em `https://github.com/user-name/repository-name/settings/hooks` e escolha Adicionar webhook.
- Em URL da carga útil, insira o valor do URL da carga útil que você anotou anteriormente.
 - Em Tipo de conteúdo, escolha `application/json`.
 - Em Segredo, insira o valor do Segredo que você anotou anteriormente.
 - Configure os eventos individuais para os quais enviarão uma carga de webhook. CodeBuild Em Para quais eventos você gostaria de acionar este webhook?, escolha Deixe-me selecionar eventos individuais e, em seguida, escolha entre os seguintes eventos: Pushes, Solicitações pull e Lançamentos. Se você quiser iniciar compilações para eventos `WORKFLOW_JOB_QUEUED`, escolha Trabalhos de fluxo de trabalho. Para saber mais sobre GitHub Actions runners, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#). Para saber mais sobre os tipos de eventos suportados pelo CodeBuild, consulte [GitHub eventos de webhook](#).
5. Escolha Adicionar webhook.

GitHub eventos de webhook

Você pode usar grupos de filtros de webhook para especificar quais eventos de GitHub webhook acionam uma compilação. Por exemplo, é possível especificar que uma compilação seja acionada somente para alterações em ramificações especificadas.

Você pode criar um ou mais grupos de filtros de webhook para especificar quais eventos de webhook acionam uma compilação. Uma compilação será acionada se algum grupo de filtros for avaliado como verdadeiro, o que ocorre quando todos os filtros no grupo são avaliados como verdadeiros. Ao criar um grupo de filtros, é necessário especificar:

Um evento

Para GitHub, você pode escolher um ou mais dos seguintes eventos:

PUSHPULL_REQUEST_CREATED,PULL_REQUEST_UPDATED,PULL_REQUEST_REOPENED,PULL_REQUEST_CLOSEDRELEASED,PRERELEASED,, WORKFLOW_JOB_QUEUED e. O tipo de evento webhook está no cabeçalho X-GitHub-Event na carga webhook. No cabeçalho X-GitHub-Event, você pode ver pull_request ou push. Para obter uma solicitação pull, o tipo de evento está no campo action da carga do evento webhook. A tabela a seguir mostra como valores de cabeçalho X-GitHub-Event e os valores do campo action da carga da solicitação pull webhook são mapeados para os tipos de eventos disponíveis.

Valor do cabeçalho X-GitHub-Event	Valor action da carga do evento Webhook	Tipo de evento
pull_request	opened	PULL_REQUEST_CREATED
pull_request	reopened	PULL_REQUEST_REOPENED
pull_request	synchronize	PULL_REQUEST_UPDATED
pull_request	closed e o campo merged é true	PULL_REQUEST_MERGED
pull_request	closed e o campo merged é false	PULL_REQUEST_CLOSED
push	n/a	PUSH

Valor do cabeçalho X-GitHub-Event	Valor action da carga do evento Webhook	Tipo de evento
release	lançado	RELEASED
release	pré-lançado	PRERELEASED
workflow_job	queued	WORKFLOW_JOB_QUEUED

 Note

O tipo de `PULL_REQUEST_REOPENED` evento pode ser usado somente com GitHub ou GitHub Enterprise Server. O tipo de `PRERELEASED` evento `RELEASED` e GitHub só pode ser usado com. Para obter mais informações sobre `WORKFLOW_JOB_QUEUED`, consulte [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

Um ou mais filtros opcionais

Use uma expressão regular para especificar um filtro. Para um evento acionar uma compilação, cada filtro do grupo associado a ele deve ser avaliado como verdadeiro.

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` no console)

Um evento de webhook aciona uma compilação quando uma ID de conta GitHub ou do GitHub Enterprise Server corresponde ao padrão de expressão regular. Esse valor é encontrado na propriedade `id` do objeto `sender` na carga webhook.

`HEAD_REF`

Um evento de webhook aciona uma compilação quando a referência head corresponde ao padrão da expressão regular (por exemplo, `refs/heads/branch-name` ou `refs/tags/tag-name`). Para um evento push, o nome de referência é encontrado na propriedade `ref` da carga webhook. Para eventos de solicitações pull, o nome da ramificação é encontrado na propriedade `ref` do objeto `head` na carga webhook.

`BASE_REF`

Um evento de webhook aciona uma compilação quando a referência base corresponde ao padrão da expressão regular (por exemplo, `refs/heads/branch-name`). Um filtro

`BASE_REF` pode ser usado apenas com eventos de solicitação pull. O nome da ramificação é encontrado na propriedade `ref` do objeto base na carga webhook.

`FILE_PATH`

Um evento de webhook aciona uma compilação quando o caminho de um arquivo alterado corresponde ao padrão das expressões regulares. Um `FILE_PATH` filtro pode ser usado com eventos GitHub push e pull request e eventos push do GitHub Enterprise Server. Ele não pode ser usado com eventos de pull request do GitHub Enterprise Server.

`COMMIT_MESSAGE`

Um webhook aciona uma compilação quando a mensagem de confirmação head corresponde à expressão padrão. Um `COMMIT_MESSAGE` filtro pode ser usado com eventos GitHub push e pull request e eventos push do GitHub Enterprise Server. Ele não pode ser usado com eventos de pull request do GitHub Enterprise Server.

`TAG_NAME`

Um webhook aciona uma compilação quando o nome da tag da versão corresponde ao padrão de expressão regular. Um `TAG_NAME` filtro pode ser usado com eventos de GitHub solicitação lançados e pré-lançados.

`RELEASE_NAME`

Um webhook aciona uma compilação quando o nome da versão corresponde ao padrão da expressão regular. Um `RELEASE_NAME` filtro pode ser usado com eventos de GitHub solicitação lançados e pré-lançados.

`REPOSITORY_NAME`

Um webhook aciona uma compilação quando o nome do repositório corresponde ao padrão da expressão regular. Um `REPOSITORY_NAME` filtro só pode ser usado com GitHub webhooks globais ou organizacionais.

`ORGANIZATION_NAME`

Um webhook aciona uma compilação quando o nome da organização corresponde ao padrão de expressão regular. Um `ORGANIZATION_NAME` filtro só pode ser usado com GitHub webhooks globais.

WORKFLOW_NAME

Um webhook aciona uma compilação quando o nome do fluxo de trabalho corresponde ao padrão da expressão regular. Um WORKFLOW_NAME filtro pode ser usado com eventos de solicitação em fila de tarefas do fluxo de trabalho do GitHub Actions.

Note

Você pode encontrar a carga útil do webhook nas configurações do webhook do seu repositório. GitHub

Tópicos

- [Filtrar eventos de GitHub webhook \(console\)](#)
- [Filtrar eventos de GitHub webhook \(SDK\)](#)
- [Filtrar eventos de GitHub webhook \(\)AWS CloudFormation](#)

Filtrar eventos de GitHub webhook (console)

Use as instruções a seguir para filtrar eventos de GitHub webhook usando o AWS Management Console. Para obter mais informações sobre eventos de GitHub webhook, consulte [GitHub eventos de webhook](#).

Em Eventos de webhook da origem principal, selecione o seguinte. Essa seção só está disponível quando você escolhe Repositório em minha GitHub conta para o repositório de origem.

1. Selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório) ao criar seu projeto.
2. Em Event type (Tipo de evento), escolha um ou mais eventos.
3. Para filtrar quando um evento aciona uma compilação, em Start a build under these conditions (Iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
4. Para filtrar quando um evento não é acionado, em Don't start a build under these conditions (Não iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
5. Escolha Adicionar grupo de filtros para adicionar outro grupo de filtros, se necessário.

Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [WebhookFilter](#) na Referência da API do AWS CodeBuild .

Neste exemplo, um grupo de filtros de webhook aciona uma compilação apenas para solicitações pull:

Filter group 1 Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X PULL_REQUEST_UPDATED X
PULL_REQUEST_REOPENED X PULL_REQUEST_MERGED X
PULL_REQUEST_CLOSED X

☐ Start a build under these conditions - *optional*

☐ Don't start a build under these conditions - *optional*

Usando um exemplo de dois grupos de filtros de webhook, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas, atualizadas ou reabertas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/branch1$`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_REOPENED ✕

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*

^refs/heads/branch1\$

BASE_REF - *optional*

^refs/heads/main\$

FILE_PATH - *optional*COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*

^refs/heads/branch1\$

BASE_REF - *optional*FILE_PATH - *optional*COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para todas as solicitações, com exceção de eventos de tag.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_REOPENED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

Neste exemplo, um grupo de filtros de webhook acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular `^buildspec.*` forem alterados.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando os arquivos são alterados nas pastas `src` ou `test`.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando uma alteração é feita por um usuário especificado GitHub ou do GitHub Enterprise Server com uma ID de conta que corresponda à expressão regular. `actor-account-id`

 Note

Para obter informações sobre como encontrar o ID GitHub da sua conta <https://api.github.com/users/user-name>, consulte Onde *user-name* está seu nome de GitHub usuário.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

Remove

► Don't start a build under these conditions - optional

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para um evento de push quando a mensagem de confirmação head corresponde à expressão regular `\[CodeBuild\]`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*COMMIT_MESSAGE -
optional

\[CodeBuild\]

► Don't start a build under these conditions

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente para eventos de trabalho do fluxo de trabalho do GitHub Actions.

 Note

CodeBuild só processará trabalhos do fluxo de trabalho do GitHub Actions se um webhook tiver grupos de filtros contendo o filtro de eventos `WORKFLOW_JOB_QUEUED`.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

► Start a build under these conditions - *optional*► Don't start a build under these conditions - *optional*

Neste exemplo, um grupo de filtros do webhook aciona uma compilação para um nome de fluxo de trabalho corresponde à expressão regular `CI-CodeBuild`.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - *optional*

Filtrar eventos de GitHub webhook (SDK)

Para usar o AWS CodeBuild SDK para filtrar eventos de webhook, use o `filterGroups` campo na sintaxe de solicitação dos métodos da API `CreateWebhook` ou `UpdateWebhook`. Para obter mais informações, consulte [WebhookFilter](#) na Referência de APIs do CodeBuild .

Para obter mais informações sobre eventos de GitHub webhook, consulte [GitHub eventos de webhook](#).

Para criar um filtro de webhook que acione uma compilação apenas para solicitações pull, insira o seguinte na sintaxe de solicitação:

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
```

```
    }
  ]
}
```

Para criar um filtro de webhook que acione uma compilação apenas para ramificações especificadas, use o parâmetro `pattern` para especificar uma expressão regular a fim de filtrar os nomes da ramificação. Usando um exemplo de dois grupos de filtros, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas, atualizadas ou reabertas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/myBranch$`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

```
]
```

Você pode usar o parâmetro `excludeMatchedPattern` para especificar quais eventos não acionam uma compilação. Por exemplo, uma compilação é acionada para todas as solicitações, com exceção de eventos de tag.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Você pode criar um filtro que acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular no argumento `pattern` forem alterados. Neste exemplo, o grupo de filtros especifica que uma compilação será acionada apenas quando arquivos com um nome que corresponda à expressão regular `^buildspec.*` forem alterados.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^buildspec.*"  
    }  
  ]  
]
```

Neste exemplo, um grupo de filtros especifica que uma compilação seja acionada somente quando os arquivos são alterados nas pastas `src` ou `test`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^src/.+|^test/.+"  
    }  
  ]  
]
```

Você pode criar um filtro que aciona uma compilação somente quando uma alteração é feita por um usuário especificado GitHub ou do GitHub Enterprise Server com ID da conta. `actor-account-id`

 Note

Para obter informações sobre como encontrar o ID GitHub da sua conta <https://api.github.com/users/user-name>, consulte Onde *user-name* está seu nome de GitHub usuário.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",  
      "pattern": "actor-account-id"  
    }  
  ]  
]
```

É possível criar um filtro que aciona uma compilação somente quando a mensagem de confirmação head corresponde à expressão regular no argumento padrão. Neste exemplo, o grupo de filtros

especifica que uma compilação é acionada somente quando a mensagem de confirmação head do evento de push corresponde à expressão regular `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\[CodeBuild\  
  ]  
]
```

Para criar um filtro de webhook que acione uma compilação somente para trabalhos de fluxo de trabalho do GitHub Actions, insira o seguinte na sintaxe da solicitação:

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "WORKFLOW_JOB_QUEUED"  
    }  
  ]  
]
```

Filtrar eventos de GitHub webhook (AWS CloudFormation)

Para usar um AWS CloudFormation modelo para filtrar eventos de webhook, use a `FilterGroups` propriedade do AWS CodeBuild projeto.

Para obter mais informações sobre eventos de GitHub webhook, consulte [GitHub eventos de webhook](#).

A seguinte parte formatada em YAML de um modelo do AWS CloudFormation cria dois grupos de filtros. Juntos, eles acionarão uma compilação quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica que as pull requests são criadas ou atualizadas em ramificações com nomes de referência do Git que correspondem à expressão regular `^refs/heads/main$` expressão regular de GitHub um usuário que não tem ID de conta. 12345
- O segundo grupo de filtros especifica que as solicitações push sejam criadas em arquivos com nomes que correspondem à expressão regular `README`, em ramificações com nomes de referência do Git que correspondem à expressão regular `^refs/heads/.*`.
- O terceiro grupo de filtros especifica uma solicitação push com uma mensagem de confirmação head correspondente à expressão regular `\[CodeBuild\]`.
- O quarto grupo de filtros especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions com um nome de fluxo de trabalho correspondente à expressão `\[CI-CodeBuild\]` regular.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
        - Type: ACTOR_ACCOUNT_ID
          Pattern: 12345
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: HEAD_REF
```

```
Pattern: ^refs/heads/.  
- Type: FILE_PATH  
  Pattern: README  
  ExcludeMatchedPattern: true  
- - Type: EVENT  
  Pattern: PUSH  
- Type: COMMIT_MESSAGE  
  Pattern: \[CodeBuild\  
- Type: FILE_PATH  
  Pattern: ^src/.+|^test/.+  
- - Type: EVENT  
  Pattern: WORKFLOW_JOB_QUEUED  
- Type: WORKFLOW_NAME  
  Pattern: \[CI-CodeBuild\  

```

GitLab webhooks em grupo

Você pode usar webhooks de CodeBuild GitLab grupo para iniciar compilações em eventos de webhook de qualquer repositório dentro de um grupo. GitLab Os webhooks de grupo funcionam com qualquer um dos tipos de eventos de GitLab webhook existentes e podem ser configurados adicionando uma configuração de escopo ao criar um webhook. CodeBuild Você também pode usar webhooks de grupo para [configurar GitLab executores auto-hospedados a fim de receber WORKFLOW_JOB_QUEUED eventos de vários repositórios CodeBuild em](#) um único projeto.

Tópicos

- [Configurar um GitLab webhook em grupo](#)
- [Filtrar eventos de webhook de GitLab grupo \(console\)](#)
- [Filtrar eventos de webhook em GitLab grupo \(\)AWS CloudFormation](#)

Configurar um GitLab webhook em grupo

As etapas de alto nível para configurar um GitLab webhook em grupo são as seguintes. Para obter mais informações sobre GitLab webhooks de grupo, consulte. [GitLab webhooks em grupo](#)

1. Defina o local de origem do seu projeto como CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION.
2. Na configuração do escopo do webhook, defina o escopo como GITLAB_GROUP.
3. Especifique um nome como parte da configuração do escopo do webhook. Para webhooks de grupo, esse é o nome do grupo.

 Note

Se o tipo de origem do projeto for `GITLAB_SELF_MANAGED`, você também precisará especificar um domínio como parte da configuração do escopo do webhook.

4. (Opcional) Se você quiser receber apenas eventos de webhook para repositórios específicos em sua organização ou empresa, você pode especificar `REPOSITORY_NAME` como filtro ao criar o webhook.
5. Ao criar um webhook de grupo, certifique-se de que ele CodeBuild tenha permissões para criar webhooks em nível de grupo. GitLab Para fazer isso, você pode CodeBuild OAuth usar CodeConnections. Para obter mais informações, consulte [GitLab acesso em CodeBuild](#).

Observe que os webhooks de grupo funcionam com qualquer um dos tipos de eventos de GitLab webhook existentes.

Filtrar eventos de webhook de GitLab grupo (console)

Ao criar um GitLab projeto por meio do console, selecione as seguintes opções para criar um webhook de GitLab grupo dentro do projeto. Para obter mais informações sobre GitLab webhooks de grupo, consulte. [GitLab webhooks em grupo](#)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
 - Em Origem:
 - Em Provedor de origem, escolha GitLabGitLabAutogerenciado.
 - Em Repositório, escolha webhook com GitLabescopo definido.

O GitLab repositório será automaticamente definido como `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, que é o local de origem necessário para webhooks de grupo.

 Note

Ao usar webhooks de grupo, verifique se ele CodeBuild tem permissões para criar webhooks em nível de grupo. GitLab Se você estiver usando uma [OAuthconexão](#)

[existente](#), talvez seja necessário regenerar a conexão para conceder CodeBuild essa permissão.

Source

[Add source](#)

Source 1 - Primary

Source provider

GitLab

Credential



Default source credential

Use your account's default source credential to apply to all projects



Custom source credential

Use a custom source credential to override your account's default settings

✔ Successfully connected through CodeConnections - [open resource](#)

[Manage default source credential](#)

Repository



Repository in my GitLab account



GitLab scoped webhook

Repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

- Em Eventos de webhook da origem principal:
- Em Nome do grupo, insira o nome do grupo.

Se o tipo de origem do projeto for `GITLAB_SELF_MANAGED`, você também precisará especificar um domínio como parte da configuração do grupo de webhook. Por exemplo, se o URL do grupo for **`https://domain.com/group/group-name`**, o domínio será **`https://domain.com`**.



Note

Não é possível alterar o nome depois que o webhook é criado. Para alterar o nome, você pode excluir e recriar o webhook. Se quiser remover completamente o webhook, você também pode atualizar o local de origem do projeto em um GitLab repositório.

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#) ☒ Rebuild every time a code change is pushed to this repository

Group name

Your GitLab group name.

Build type



Single build

Triggers single build



Batch build

Triggers multiple builds as single execution

[► Additional configuration](#)

- (Opcional) Nos grupos de filtros de eventos de Webhook, você pode especificar quais [eventos você gostaria de acionar uma nova compilação](#). Você também pode especificar `REPOSITORY_NAME` como filtro para acionar somente compilações em eventos de webhook de repositórios específicos.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Você também pode definir o tipo de evento WORKFLOW_JOB_QUEUED para configurar GitLab corretores auto-hospedados. Para obter mais informações, consulte [GitLab Corredores autogerenciados em AWS CodeBuild](#).

- Continue com os valores padrão e escolha Criar projeto de compilação.

Filtrar eventos de webhook em GitLab grupo ()AWS CloudFormation

Para usar um AWS CloudFormation modelo para filtrar eventos de webhook de grupo, use a ScopeConfiguration propriedade do AWS CodeBuild projeto. Para obter mais informações sobre GitLab webhooks de grupo, consulte. [GitLab webhooks em grupo](#)

A seguinte parte formatada em YAML de um AWS CloudFormation modelo cria quatro grupos de filtros. Juntos, eles acionarão uma compilação quando um ou todos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica que as pull requests são criadas ou atualizadas em ramificações com nomes de referência do Git que correspondem à `^refs/heads/main$` expressão regular de GitLab um usuário que não tem ID de conta. 12345

- O segundo grupo de filtros especifica que as solicitações push sejam criadas em arquivos com nomes que correspondem à expressão regular `READ_ME`, em ramificações com nomes de referência do Git que correspondem à expressão regular `^refs/heads/.*`.
- O terceiro grupo de filtros especifica uma solicitação push com uma mensagem de confirmação head correspondente à expressão regular `\[CodeBuild\]`.
- O quarto grupo de filtros especifica um nome de GitLab CI/CD pipeline job request with a CI/CD pipeline que corresponde à expressão `\[CI-CodeBuild\]` regular.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
      - Type: ACTOR_ACCOUNT_ID
        Pattern: 12345
        ExcludeMatchedPattern: true
      - - Type: EVENT
        Pattern: PUSH
      - Type: HEAD_REF
        Pattern: ^refs/heads/.*      - Type: FILE_PATH
```

```
Pattern: READ_ME
ExcludeMatchedPattern: true
- - Type: EVENT
Pattern: PUSH
- Type: COMMIT_MESSAGE
Pattern: \[CodeBuild\]
- Type: FILE_PATH
Pattern: ^src/.+|^test/.+
- - Type: EVENT
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]
```

GitLab webhooks manuais

Você pode configurar GitLab webhooks manuais para evitar a tentativa automática CodeBuild de criar um webhook nele. GitLab CodeBuild retorna um URL de carga útil como parte da chamada para criar o webhook e pode ser usado para criar manualmente o webhook dentro dele. Mesmo que não CodeBuild esteja na lista de permissões para criar um webhook em sua GitLab conta, você ainda pode criar manualmente um webhook para seu projeto de compilação.

Use o procedimento a seguir para criar um webhook GitLab manual.

Para criar um GitLab webhook manual

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
 - Em Origem:
 - Em Source provider, escolha GitLab.
 - Em Repositório, escolha Repositório na minha GitLab conta.
 - Em URL do repositório, insira **`https://gitlab.com/user-name/repository-name`**.
 - Em Eventos de webhook da origem principal:
 - Em Webhook - opcional, selecione Recompilar toda vez que uma alteração de código for enviada para este repositório.
 - Escolha Configuração adicional e, para Criação manual - opcional, escolha Criar manualmente um webhook para este repositório no GitLab console. .

3. Continue com os valores padrão e escolha Criar projeto de compilação. Anote o URL da carga útil e os valores do Segredo, pois serão usados posteriormente.
4. Abra o GitLab console em <https://gitlab.com/user-name/repository-name/-/hooks> e escolha Adicionar novo webhook.
 - Em URL, insira o valor do URL de carga que você anotou anteriormente.
 - Em Token secreto, insira o valor secreto que você anotou anteriormente.
 - Configure os eventos individuais para os quais enviarão uma carga de webhook. CodeBuild Para Trigger, escolha entre os seguintes eventos: eventos push, eventos de solicitação de mesclagem, eventos de lançamentos e eventos de Job. Para saber mais sobre os tipos de eventos suportados pelo CodeBuild, consulte [GitLab eventos de webhook](#).
5. Escolha Adicionar webhook.

GitLab eventos de webhook

Você pode usar grupos de filtros de webhook para especificar quais eventos de GitLab webhook acionam uma compilação. Por exemplo, é possível especificar que uma compilação seja acionada somente para alterações em ramificações especificadas.

Você pode criar um ou mais grupos de filtros de webhook para especificar quais eventos de webhook acionam uma compilação. Uma compilação será acionada se algum grupo de filtros for avaliado como verdadeiro, o que ocorre quando todos os filtros no grupo são avaliados como verdadeiros. Ao criar um grupo de filtros, é necessário especificar:

Um evento

Para GitLab, você pode escolher um ou mais dos seguintes eventos:

PUSH_PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED, PULL_REQUEST_REOPENED, PULL_REQUEST_CLOSED, RELEASED, WORKFLOW_JOB_QUEUED e.

O tipo de evento do webhook está em seu cabeçalho no campo X-GitLab-Event. A tabela a seguir mostra como valores de cabeçalho X-GitLab-Event são mapeados para os tipos de eventos. Para o evento de webhook Merge Request Hook, a `object_attributes.action` da carga útil conterá informações adicionais sobre o tipo de solicitação de mesclagem.

Valor do cabeçalho X- GitLab-Event	object_attributes .action	Tipo de evento
Push Hook	N/D	PUSH
Merge Request Hook	aberto	PULL_REQUEST_CREATED
Merge Request Hook	atualizar	PULL_REQUEST_UPDATED
Merge Request Hook	merge	PULL_REQUEST_MERGED
Merge Request Hook	reabrir	PULL_REQUEST_REOPENED
Merge Request Hook	feche	PULL_REQUEST_CLOSED
Release Hook	criar, atualizar	RELEASED
Job Hook	N/D	WORKFLOW_JOB_QUEUED

Para `PULL_REQUEST_MERGED`, se uma solicitação pull for mesclada com a estratégia squash e a ramificação da solicitação pull for fechada, a confirmação original da solicitação pull deixará de existir. Nesse caso, a variável de ambiente `CODEBUILD_WEBHOOK_MERGE_COMMIT` contém o identificador da confirmação de mesclagem comprimida.

Um ou mais filtros opcionais

Use uma expressão regular para especificar um filtro. Para um evento acionar uma compilação, cada filtro do grupo associado a ele deve ser avaliado como verdadeiro.

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` no console)

Um evento de webhook aciona uma compilação quando o ID da GitLab conta corresponde ao padrão de expressão regular. Esse valor é exibido na propriedade `account_id` do objeto `actor` na carga de filtro webhook.

`HEAD_REF`

Um evento de webhook aciona uma compilação quando a referência head corresponde ao padrão da expressão regular (por exemplo, `refs/heads/branch-name` e `refs/tags/tag-name`). Um filtro `HEAD_REF` avalia o nome de referência do Git para a ramificação ou tag.

O nome de ramificação ou de tag é exibido no campo `name` do objeto `new` no objeto `push` da carga webhook. Para eventos de solicitação `pull`, o nome da ramificação é exibido no campo `name` no objeto `branch` do objeto `source` na carga webhook.

BASE_REF

Um evento de webhook aciona uma compilação quando a referência base corresponde ao padrão da expressão regular. Um filtro `BASE_REF` funciona apenas com eventos de solicitação `pull` (por exemplo, `refs/heads/branch-name`). Um filtro `BASE_REF` avalia o nome de referência do Git para a ramificação. O nome da ramificação é exibido no campo `name` do objeto `branch` no objeto `destination` na carga webhook.

FILE_PATH

Um evento de webhook aciona uma compilação quando o caminho de um arquivo alterado corresponde ao padrão da expressão regular.

COMMIT_MESSAGE

Um webhook aciona uma compilação quando a mensagem de confirmação `head` corresponde à expressão padrão.

WORKFLOW_NAME

Um webhook aciona uma compilação quando o nome do fluxo de trabalho corresponde ao padrão da expressão regular.

Note

Você pode encontrar a carga útil do webhook nas configurações do webhook do seu repositório. GitLab

Tópicos

- [Filtrar eventos de GitLab webhook \(console\)](#)
- [Filtrar eventos de GitLab webhook \(SDK\)](#)
- [Filtrar eventos de GitLab webhook \(AWS CloudFormation\)](#)

Filtrar eventos de GitLab webhook (console)

Use as instruções a seguir para usar o AWS Management Console para filtrar eventos de webhook. Para obter mais informações sobre eventos de GitLab webhook, consulte [GitLab eventos de webhook](#).

1. Selecione Rebuild every time a code change is pushed to this repository (Recompilar toda vez que uma alteração de código for enviada para este repositório) ao criar seu projeto.
2. Em Event type (Tipo de evento), escolha um ou mais eventos.
3. Para filtrar quando um evento aciona uma compilação, em Start a build under these conditions (Iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
4. Para filtrar quando um evento não é acionado, em Don't start a build under these conditions (Não iniciar uma compilação sob estas condições), adicione um ou mais filtros opcionais.
5. Escolha Add filter group (Adicionar grupo de filtros) para adicionar outro grupo de filtros.

Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [WebhookFilter](#) na Referência da API do AWS CodeBuild .

Neste exemplo, um grupo de filtros de webhook aciona uma compilação apenas para solicitações pull:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

► Start a build under these conditions - *optional*

► Don't start a build under these conditions - *optional*

Usando um exemplo de dois grupos de filtros, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas ou atualizadas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/branch1!`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/branch1$`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ×PULL_REQUEST_UPDATED ×

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Filter 2

Type

Pattern

[Remove](#)

► **Don't start a build under these conditions - optional**

Filter group 2

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ×

Filter 1

Type

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para todas as solicitações, com exceção de eventos de tag.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_MERGED](#) ✕

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

Neste exemplo, um grupo de filtros de webhook acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular `^buildspec.*` forem alterados.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - *optional*

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando os arquivos são alterados nas pastas `src` ou `test`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

FILE_PATH

Pattern

^src/.+|^test/.+

Remove

► Don't start a build under these conditions - optional

Neste exemplo, um grupo de filtros de webhook aciona uma compilação somente quando uma alteração é feita por um GitLab usuário que não tem um ID de conta que corresponda à expressão regular. `actor-account-id`

Note

Para obter informações sobre como encontrar o ID GitLab da sua conta <https://api.github.com/users/user-name>, consulte Onde *user-name* está seu nome de GitLab usuário.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) [×](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Neste exemplo, um grupo de filtros de webhook aciona uma compilação para um evento de push quando a mensagem de confirmação head corresponde à expressão regular `\[CodeBuild\]`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) [×](#)

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► **Don't start a build under these conditions - optional**

Filtrar eventos de GitLab webhook (SDK)

Para usar o AWS CodeBuild SDK para filtrar eventos de webhook, use o `filterGroups` campo na sintaxe de solicitação dos métodos da API `CreateWebhook` ou `UpdateWebhook`. Para obter mais informações, consulte [WebhookFilter](#) na Referência de APIs do CodeBuild .

Para obter mais informações sobre eventos de GitLab webhook, consulte [GitLab eventos de webhook](#).

Para criar um filtro de webhook que acione uma compilação apenas para solicitações pull, insira o seguinte na sintaxe de solicitação:

```
"filterGroups": [  
  [  
    {
```

```
    "type": "EVENT",
    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED"
  }
]
```

Para criar um filtro de webhook que acione uma compilação apenas para ramificações especificadas, use o parâmetro `pattern` para especificar uma expressão regular a fim de filtrar os nomes da ramificação. Usando um exemplo de dois grupos de filtros, uma compilação será acionada quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica as solicitações pull que são criadas ou atualizadas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/main$` e referências head que correspondam a `^refs/heads/myBranch$`.
- O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

```
]
]
```

Você pode usar o parâmetro `excludeMatchedPattern` para especificar quais eventos não acionam uma compilação. Neste exemplo, um compilação é acionada para todas as solicitações, com exceção de eventos de tag.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/tags/.*",
      "excludeMatchedPattern": true
    }
  ]
]
```

Você pode criar um filtro que aciona uma compilação somente quando uma alteração é feita por um GitLab usuário com ID da conta. `actor-account-id`

Note

Para obter informações sobre como encontrar o ID GitLab da sua conta <https://api.github.com/users/user-name>, consulte Onde *user-name* está seu nome de GitLab usuário.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
```

```
    "pattern": "actor-account-id"
  }
]
]
```

Você pode criar um filtro que acionará uma compilação apenas quando arquivos com nomes que correspondam à expressão regular no argumento `pattern` forem alterados. Neste exemplo, o grupo de filtros especifica que uma compilação será acionada apenas quando arquivos com um nome que corresponda à expressão regular `^buildspec.*` forem alterados.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

Neste exemplo, um grupo de filtros especifica que uma compilação seja acionada somente quando os arquivos são alterados nas pastas `src` ou `test`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

É possível criar um filtro que aciona uma compilação somente quando a mensagem de confirmação `head` corresponde à expressão regular no argumento padrão. Neste exemplo, o grupo de filtros

especifica que uma compilação é acionada somente quando a mensagem de confirmação head do evento de push corresponde à expressão regular `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\[CodeBuild\  
  ]  
]
```

Filtrar eventos de GitLab webhook ()AWS CloudFormation

Para usar um AWS CloudFormation modelo para filtrar eventos de webhook, use a `FilterGroups` propriedade do AWS CodeBuild projeto. Para obter mais informações sobre eventos de GitLab webhook, consulte [GitLab eventos de webhook](#).

A seguinte parte formatada em YAML de um modelo do AWS CloudFormation cria dois grupos de filtros. Juntos, eles acionarão uma compilação quando um ou ambos forem avaliados como verdadeiro:

- O primeiro grupo de filtros especifica que as pull requests são criadas ou atualizadas em ramificações com nomes de referência do Git que correspondem à expressão regular de GitLab `^refs/heads/main$` um usuário que não tem ID de conta. 12345
- O segundo grupo de filtros especifica solicitações push que são criadas em ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/.*`.
- O terceiro grupo de filtros especifica uma solicitação push com uma mensagem de confirmação head correspondente à expressão regular `\[CodeBuild\]`.
- O quarto grupo de filtros especifica uma solicitação de trabalho do fluxo de trabalho do GitHub Actions com um nome de fluxo de trabalho correspondente à expressão `\[CI-CodeBuild\
regular.`

```
CodeBuildProject:  
  Type: AWS::CodeBuild::Project
```

```
Properties:
  Name: MyProject
  ServiceRole: service-role
  Artifacts:
    Type: NO_ARTIFACTS
  Environment:
    Type: LINUX_CONTAINER
    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/standard:5.0
  Source:
    Type: GITLAB
    Location: source-location
  Triggers:
    Webhook: true
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
      - Type: ACTOR_ACCOUNT_ID
        Pattern: 12345
        ExcludeMatchedPattern: true
      - - Type: EVENT
        Pattern: PUSH
      - Type: HEAD_REF
        Pattern: ^refs/heads/. *
      - - Type: EVENT
        Pattern: PUSH
      - Type: COMMIT_MESSAGE
        Pattern: \[CodeBuild\]
      - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
      - Type: WORKFLOW_NAME
        Pattern: \[CI-CodeBuild\]
```

Webhooks manuais do Buildkite

Atualmente, CodeBuild exige que todos os webhooks do Buildkite sejam criados manualmente. CodeBuild retorna uma URL de carga útil como parte da chamada para criar o webhook, que pode ser usado para criar manualmente o webhook no Buildkite.

Use o procedimento a seguir para criar um webhook manual do Buildkite.

Para criar um CodeBuild projeto com um webhook

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Crie um projeto de compilação. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
3. Em Configuração do projeto, escolha Projeto Runner.

Em Runner:

- Para o provedor Runner, escolha Buildkite.
 - Para token de agente Buildkite, escolha Criar um novo token de agente usando a página de criação secreta. Você será solicitado a criar um novo segredo no AWS Secrets Manager com um valor secreto igual ao token do agente Buildkite que você gerou acima.
 - (Opcional) Se você quiser usar credenciais CodeBuild gerenciadas para seu trabalho, selecione o provedor do repositório de origem do seu trabalho nas opções de credenciais de origem do Buildkite e verifique se as credenciais estão configuradas para sua conta. Além disso, verifique se o pipeline do Buildkite usa o Checkout usando HTTPS.
4. • Em Ambiente:
 - Escolha uma imagem de ambiente e computação compatíveis. Observe que você tem a opção de substituir as configurações de imagem e instância usando um rótulo no seu fluxo de trabalho de GitHub ações YAML. Para ter mais informações, consulte [Etapa 2: atualizar seu fluxo de trabalho de GitHub ações YAML](#)
 - Em Buildspec:
 - Observe que o buildspec será ignorado, a menos que `buildspec-override:true` seja adicionado como rótulo. Em vez disso, o CodeBuild substituirá para usar comandos que configurarão o executor auto-hospedado.
 5. Continue com os valores padrão e escolha Criar projeto de compilação.
 6. Salve o URL da carga útil e os valores secretos no pop-up Criar Webhook. Siga as instruções no pop-up para criar um novo webhook da organização Buildkite.

Veja os detalhes de um projeto de construção em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para ver os detalhes de um projeto de compilação em CodeBuild.

Tópicos

- [Visualizar detalhes de um projeto de compilação \(console\)](#)
- [Visualizar detalhes de um projeto de compilação \(AWS CLI\)](#)
- [Exibir detalhes de um projeto de construção \(AWS SDKs\)](#)

Visualizar detalhes de um projeto de compilação (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.

Note

Por padrão, somente os 10 projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Projetos por página ou use as setas para voltar e avançar.

3. Na lista de projetos de compilação, na coluna Name (Nome), selecione o link correspondente ao nome do projeto de compilação.
4. Na *project-name* página Build project:, escolha Build details.

Visualizar detalhes de um projeto de compilação (AWS CLI)

Execute o comando batch-get-projects:

```
aws codebuild batch-get-projects --names names
```

No comando anterior, substitua o seguinte espaço reservado:

- *names*: seqüência de caracteres obrigatória usada para indicar um ou mais nomes de projetos de construção para visualizar detalhes sobre. Para especificar mais de um projeto de build, separe cada nome com um espaço. Você pode especificar até 100 projetos de build. Para obter uma lista de projetos de compilação, consulte [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#).

Por exemplo, se você executar este comando:

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2
my-other-demo-project
```

Um resultado semelhante ao seguinte pode ser exibido na saída. Elipses (...) são usadas para representar dados omitidos para agilizar.

```
{
  "projectsNotFound": [
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": codebuild-demo-project,
      ...
    },
    {
      ...
      "name": codebuild-demo-project2",
      ...
    }
  ]
}
```

Na saída anterior, a matriz `projectsNotFound` lista nomes de projetos de compilação que foram especificados, mas não encontrados. A matriz `projects` lista detalhes para cada projeto de build em que informações foram encontradas. Detalhes de projeto de build foram omitidos de saída anterior para agilizar. Para obter mais informações, consulte a saída de [Criar um projeto de compilação \(AWS CLI\)](#).

O comando `batch-get-projects` não é compatível com a filtragem de determinados valores de propriedade, mas é possível elaborar um script que enumere as propriedades de um projeto. Por exemplo, o script de shell Linux a seguir enumera os projetos na região atual da conta atual e imprime a imagem usada por cada projeto.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.
```

```

imageName=""

function getImageName(){
    local environmentValues=(${1//$'\t'/ })
    imageName=${environmentValues[1]}
}

function processProjectInfo() {
    local projectInfo=$1

    while IFS=$'\t' read -r section value; do
        if [[ "$section" == *"ENVIRONMENT"* ]]; then
            getImageName "$value"
        fi
    done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
    if [[ "$projectName" != *"PROJECTS"* ]]; then
        echo "=====

        # Get the detailed information for the project.
        projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

        processProjectInfo "$projectInfo"

        printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
    fi
done

```

Para obter mais informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Exibir detalhes de um projeto de construção (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Veja os nomes dos projetos de construção em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para ver uma lista de projetos de compilação em CodeBuild.

Tópicos

- [Visualizar uma lista de nomes de projetos de compilação \(console\)](#)
- [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#)
- [Veja uma lista de nomes de projetos de construção \(AWS SDKs\)](#)

Visualizar uma lista de nomes de projetos de compilação (console)

Você pode ver uma lista de projetos de compilação em uma AWS região no console. As informações incluem o nome, o provedor de origem, o repositório, o status de compilação mais recente e a descrição, se houver.

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.

Note

Por padrão, somente os 10 projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Projetos por página ou use as setas para voltar e avançar.

Visualizar uma lista de nomes de projetos de compilação (AWS CLI)

Execute o comando `list-projects`:

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

No comando anterior, substitua os seguintes espaços reservados:

- *sort-by*: string opcional usada para indicar o critério a ser usado para listar os nomes dos projetos de construção. Os valores válidos são:

- `CREATED_TIME`: Listar os nomes de projetos de build pela data e hora de criação do projeto.
- `LAST_MODIFIED_TIME`: Listar os nomes de projetos de build pela data e hora da última alteração de cada projeto.
- `NAME`: Listar os nomes de projetos de build pelo nome de cada projeto.
- *`sort-order`*: string opcional usada para indicar a ordem na qual listar os projetos de construção, com base em *`sort-by`*. Os valores válidos são `ASCENDING` e `DESCENDING`.
- *`next-token`*: Cadeia de caracteres opcional. Se houver mais de 100 itens na lista em uma execução anterior, somente os primeiros 100 itens serão retornados, juntamente com uma string exclusiva de próximo token. Para obter o próximo lote de itens na lista, execute o comando novamente, adicionando o próximo token à chamada. Para obter todos os itens na lista, continue executando esse comando com cada próximo token subsequente, até que não haja mais tokens.

Por exemplo, se você executar este comando:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Se você executar este comando novamente:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "projects": [
    "codebuild-demo-project100",
```

```
"codebuild-demo-project101",  
... The full list of build project names has been omitted for brevity ...  
"codebuild-demo-project122"  
]  
}
```

Veja uma lista de nomes de projetos de construção (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Incorpora em AWS CodeBuild

Uma construção representa um conjunto de ações realizadas AWS CodeBuild para criar artefatos de saída (por exemplo, um arquivo JAR) com base em um conjunto de artefatos de entrada (por exemplo, uma coleção de arquivos de classe Java).

As seguintes regras se aplicam ao executar várias compilações:

- Quando possível, as compilações são executadas simultaneamente. O número máximo compilações em execução simultaneamente pode variar. Para obter mais informações, consulte [Cotas para AWS CodeBuild](#).
- Se o projeto de compilação tiver um limite de compilação simultânea definido, as compilações vão gerar um erro se o número de compilações em execução atingir o limite de compilações simultâneas do projeto. Para obter mais informações, consulte [Habilitar limite de compilações simultâneas](#).
- Se o projeto de compilação não tiver um limite de compilação simultânea definido, as compilações serão colocadas em fila se o número de compilações em execução atingir o limite de compilações simultâneas para a plataforma e o tipo de computação. O número máximo de compilações em uma fila é cinco vezes o limite de compilações simultâneas. Para obter mais informações, consulte [Cotas para AWS CodeBuild](#).

Uma compilação em uma fila que não é iniciada após o número de minutos especificado no seu valor de tempo limite é removida da fila. O valor do tempo limite padrão é de 8 horas. Você pode substituir o tempo limite da fila de compilação por um valor entre cinco minutos e oito horas quando você executar a compilação. Para obter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#).

Não é possível prever a ordem na qual as compilações colocadas em fila começam.

Note

É possível acessar até um ano de histórico de uma compilação.

Você pode executar estas tarefas para trabalhar com builds:

Tópicos

- [Execute AWS CodeBuild compilações manualmente](#)
- [Execute compilações em computação AWS Lambda](#)
- [Executar compilações em frotas de capacidade reservada](#)
- [Executar compilações em lotes](#)
- [Execute testes paralelos em compilações em lote](#)
- [Compilações em cache para melhorar o desempenho](#)
- [Compilações de depuração AWS CodeBuild](#)
- [Excluir compilações em AWS CodeBuild](#)
- [Repita as compilações manualmente em AWS CodeBuild](#)
- [Repetir compilações automaticamente em AWS CodeBuild](#)
- [Pare de incorporar AWS CodeBuild](#)
- [Interrompa a criação de lotes AWS CodeBuild](#)
- [O gatilho AWS CodeBuild é criado automaticamente](#)
- [Veja os detalhes da construção em AWS CodeBuild](#)
- [Veja uma lista de IDs incorporações AWS CodeBuild](#)
- [Veja uma lista de compilações IDs para um projeto de compilação em AWS CodeBuild](#)

Execute AWS CodeBuild compilações manualmente

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para executar uma compilação CodeBuild.

Tópicos

- [Execute compilações localmente com o agente AWS CodeBuild](#)
- [Executar uma compilação \(console\)](#)
- [Executar uma compilação \(AWS CLI\)](#)
- [Executar uma compilação em lote \(AWS CLI\)](#)
- [Começar a executar compilações automaticamente \(AWS CLI\)](#)
- [Parar de executar compilações automaticamente \(AWS CLI\)](#)
- [Executar uma compilação \(AWS SDKs\)](#)

Execute compilações localmente com o agente AWS CodeBuild

Você pode usar o AWS CodeBuild agente para executar CodeBuild compilações em uma máquina local. Há agentes disponíveis para plataformas x86_64 e ARM.

Também é possível assinar para receber notificações quando novas versões do agente são publicadas.

Pré-requisitos

Antes de começar, será necessário fazer o seguinte:

- Instale o Git na sua máquina local.
- Instale e configure o [Docker](#) na máquina local.

Configurar a imagem de compilação

Basta configurar a imagem de compilação na primeira vez em que executar o agente ou quando a imagem for alterada.

Como configurar a imagem de compilação

1. [Se quiser usar uma imagem curada do Amazon Linux 2, você pode extraí-la do repositório CodeBuild público do Amazon ECR em https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard](https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard) com o seguinte comando:

```
$ docker pull public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0
```

Como alternativa, se você deseja usar outra imagem do Linux, execute as seguintes etapas:

- a. Clone o repositório CodeBuild de imagens:

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Mude para o diretório de imagens. Para este exemplo, use a imagem aws/codebuild/standard:5.0:

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Crie a imagem. Isso levará vários minutos.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Baixe o CodeBuild agente.

Para baixar a versão x86_64 do agente, execute o seguinte comando:

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Para baixar a versão ARM do agente, execute o seguinte comando:

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

3. O CodeBuild agente está disponível em <https://gallery.ecr.aws/codebuild/local-builds>.

A assinatura do Secure Hash Algorithm (SHA) para a versão x86_64 do agente é:

```
sha256:ccb19bdd7af94e4dc761e4c58c267e9455c28ec68d938086b4dc1cf8fe6b0940
```

A assinatura SHA para a versão ARM do agente é:

```
sha256:7d7b5d35d2ac4e062ae7ba8c662ffed15229a52d09bd0d664a7816c439679192
```

É possível usar o SHA para identificar a versão do agente. Para ver a assinatura SHA do agente, execute o seguinte comando e procure o SHA em RepoDigests:

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

Execute o CodeBuild agente

Para executar o CodeBuild agente

1. Mude para o diretório que contém a fonte do projeto de compilação.
2. Baixe o script [codebuild_build.sh](https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh).

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh
$ chmod +x codebuild_build.sh
```

3. Execute o script `codebuild_build.sh` e especifique as imagens de contêiner e o diretório de saída:

Para executar uma compilação x86_64, execute o seguinte comando:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Para executar uma compilação ARM, execute o seguinte comando:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

<container-image> Substitua pelo nome da imagem do contêiner, como `aws/codebuild/standard:5.0` ou `public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0`.

O script inicia a imagem de compilação e executa a compilação no projeto no diretório atual.

Para especificar a localização do projeto de compilação, adicione a opção `-s <build project directory>` ao comando script.

Receber notificações sobre novas versões do agente do CodeBuild

Você pode assinar as notificações do Amazon SNS para ser notificado quando novas versões do AWS CodeBuild agente forem lançadas.

Para assinar as notificações CodeBuild do agente

1. [Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home](https://console.aws.amazon.com/sns/v3/home).
2. Na barra de navegação, se ainda não estiver selecionada, altere a AWS região para Leste dos EUA (Norte da Virgínia). Você deve selecionar essa AWS região porque as notificações do Amazon SNS que você está assinando são criadas nessa região.
3. No painel de navegação, escolha **Subscriptions**.
4. Selecione **Create subscription**.
5. Em **Criar inscrição**, faça o seguinte:
 - a. Para **Topic ARN** (ARN do tópico), use o seguinte nome do recurso da Amazon (ARN):

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. Em Protocolo, escolha Email ou SMS.
- c. Em Endpoint, escolha onde deseja receber as notificações (e-mail ou SMS). Digite um e-mail, um endereço ou número de telefone, incluindo o código de área.
- d. Selecione Criar assinatura.
- e. Se você escolher E-mail, receberá um e-mail solicitando que confirme a inscrição. Siga as instruções no e-mail para concluir sua inscrição.

Se não deseja mais receber essas notificações, use o procedimento a seguir para cancelar a assinatura.

Para cancelar a assinatura das notificações do CodeBuild agente

1. [Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home](https://console.aws.amazon.com/sns/v3/home).
2. No painel de navegação, escolha Assinaturas.
3. Selecione a assinatura e, em Actions (Ações), escolha Delete subscriptions (Excluir assinaturas). Quando for solicitada sua confirmação, escolha Delete.

Executar uma compilação (console)

Para usar AWS CodePipeline para executar uma compilação com CodeBuild, pule essas etapas e siga as instruções em [Use CodeBuild com CodePipeline](#).

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Na lista de projetos de compilação, selecione o projeto de compilação.
4. É possível executar a compilação com as configurações padrão do projeto de compilação ou substituí-las somente para essa compilação.
 - a. Se você quiser executar a compilação com as configurações padrão do projeto de compilação, escolha Iniciar compilação. A compilação é iniciada imediatamente.
 - b. Se quiser substituir as configurações padrão do projeto de compilação, escolha Iniciar compilação com substituições. Na página Iniciar compilação, é possível substituir o seguinte:
 - Configuração de compilação

- Origem
- Substituições de variáveis de ambiente

Se precisar selecionar substituições mais avançadas, escolha Substituições avançadas de compilações. Nessa página, é possível substituir o seguinte:

- Configuração de compilação
- Origem
- Ambiente
- Buildspec
- Artefatos
- Logs

Depois de fazer as seleções de substituição, escolha Iniciar compilação.

Para obter informações detalhadas sobre este build, consulte [Visualizar detalhes de compilação \(console\)](#).

Executar uma compilação (AWS CLI)

Note

Para usar CodePipeline para executar uma compilação com AWS CodeBuild, pule essas etapas e siga as instruções em [Criar um pipeline que use o CodeBuild \(AWS CLI\)](#).

Para obter mais informações sobre como usar o AWS CLI with CodeBuild, consulte [Referência da linha de comando](#) o.

1. Execute o comando `start-build` de uma das seguintes maneiras:

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

Use esta maneira se você quiser executar um build que utiliza a versão mais recente do artefato de entrada de build e as configurações existentes do projeto de build.

```
aws codebuild start-build --generate-cli-skeleton
```

Use esta maneira se quiser executar uma compilação com uma versão anterior do artefato de entrada da compilação ou se quiser substituir as configurações dos artefatos de saída da compilação, as variáveis de ambiente, o buildspec ou o período de tempo limite padrão da compilação.

2. Se você executar o start-build comando com a --project-name opção, *<project-name>* substitua pelo nome do projeto de compilação e, em seguida, vá para a etapa 6 desse procedimento. Para obter uma lista de projetos de compilação, consulte [Exibir os nomes do projeto de compilação](#).
3. Se você executar o comando start-build com a opção --idempotency-token, um identificador, ou token, exclusivo que diferencia maiúsculas e minúsculas é incluído na solicitação start-build. O token é válido por 5 minutos após a solicitação. Se você repetir a start-build solicitação com o mesmo token, mas alterar um parâmetro, CodeBuild retornará um erro de incompatibilidade de parâmetros.
4. Se você executar o comando start-build com a opção --generate-cli-skeleton, os dados formatados JSON serão exibidos na saída. Copie os dados para um arquivo (por exemplo, *start-build.json*) em um local no computador local ou na instância em que o AWS CLI está instalado. Altere os dados copiados para corresponder ao seguinte formato, e salve os resultados:

```
{
  "projectName": "projectName",
  "sourceVersion": "sourceVersion",
  "artifactsOverride": {
    "type": "type",
    "location": "location",
    "path": "path",
    "namespaceType": "namespaceType",
    "name": "artifactsOverride-name",
    "packaging": "packaging"
  },
  "buildspecOverride": "buildspecOverride",
  "cacheOverride": {
    "location": "cacheOverride-location",
    "type": "cacheOverride-type"
  },
  "certificateOverride": "certificateOverride",
```

```
"computeTypeOverride": "computeTypeOverride",
"environmentTypeOverride": "environmentTypeOverride",
"environmentVariablesOverride": {
  "name": "environmentVariablesOverride-name",
  "value": "environmentVariablesValue",
  "type": "environmentVariablesOverride-type"
},
"gitCloneDepthOverride": "gitCloneDepthOverride",
"imageOverride": "imageOverride",
"idempotencyToken": "idempotencyToken",
"insecureSslOverride": "insecureSslOverride",
"privilegedModeOverride": "privilegedModeOverride",
"queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
"reportBuildStatusOverride": "reportBuildStatusOverride",
"timeoutInMinutesOverride": "timeoutInMinutesOverride",
"sourceAuthOverride": "sourceAuthOverride",
"sourceLocationOverride": "sourceLocationOverride",
"serviceRoleOverride": "serviceRoleOverride",
"sourceTypeOverride": "sourceTypeOverride"
}
```

Substitua os seguintes espaços reservados:

- **projectName**: Cadeia de caracteres obrigatória. O nome de projeto de build a ser usado neste build.
- **sourceVersion**: Cadeia de caracteres opcional. Uma versão do código-fonte a ser compilado, como se segue:
 - Para o Amazon S3, o ID de versão correspondente à versão do arquivo ZIP de entrada que você quer compilar. Se não **sourceVersion** for especificado, a versão mais recente será usada.
 - Para CodeCommit, o ID do commit que corresponde à versão do código-fonte que você deseja criar. Se não **sourceVersion** for especificado, o ID de confirmação HEAD da ramificação padrão será usado. (Você não pode especificar um nome de tag para **sourceVersion**, mas pode especificar o ID de confirmação da tag.)
 - Para GitHub, o ID do commit, o ID do pull request, o nome da ramificação ou o nome da tag que corresponde à versão do código-fonte que você deseja criar. Caso seja especificado, um ID de solicitação deve usar o formato `pr/pull-request-ID` (por exemplo, `pr/25`). Se um nome de ramificação for especificado, o ID de confirmação HEAD da ramificação

será usado. Se não *sourceVersion* for especificado, o ID de confirmação HEAD da ramificação padrão será usado.

- Para o Bitbucket, o ID de confirmação, o nome da ramificação ou o nome da tag que corresponda à versão do código-fonte que você deseja compilar. Se um nome de ramificação for especificado, o ID de confirmação HEAD da ramificação será usado. Se não *sourceVersion* for especificado, o ID de confirmação HEAD da ramificação padrão será usado.
- Os seguintes espaços reservados são para: *artifactsOverride*.
 - *type*: opcional. O tipo de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
 - *location*: opcional. O local de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
 - *path*: opcional. O caminho de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
 - *namespaceType*: opcional. O tipo de caminho de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
 - *name*: opcional. O nome de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
 - *packaging*: opcional. O tipo de empacotamento de artefato de saída de build que sobrescreve neste build aquele definido no projeto de build.
- *buildspecOverride*: opcional. Uma declaração de buildspec que sobrescreve para esta compilação aquela definida no projeto de compilação. Se estiver definido, esse valor poderá ser uma definição de buildspec em linha ou o caminho para um arquivo buildspec alternativo relativo ao valor da variável de ambiente `CODEBUILD_SRC_DIR` interna ou o caminho para um bucket do S3. O bucket do S3 deve estar na mesma região da AWS que o projeto de compilação. Especifique o arquivo buildspec usando seu ARN (por exemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Se esse valor não for fornecido ou for definido como uma string vazia, o código-fonte deverá conter um arquivo `buildspec.yml` em seu diretório raiz. Para obter mais informações, consulte [Nome do arquivo buildspec e local de armazenamento](#).
- Os seguintes espaços reservados são para: *cacheOverride*.
 - *cacheOverride-location*: opcional. O local de um objeto ProjectCache desta compilação que substitui o objeto ProjectCache especificado no projeto de compilação.

`cacheOverride` é opcional e utiliza um objeto `ProjectCache`. `location` é necessário em um objeto `ProjectCache`.

- **`cacheOverride-type`**: opcional. O tipo de um objeto `ProjectCache` desta compilação que substitui o objeto `ProjectCache` especificado no projeto de compilação. `cacheOverride` é opcional e utiliza um objeto `ProjectCache`. `type` é necessário em um objeto `ProjectCache`.
- **`certificateOverride`**: opcional. O nome de um certificado para esse build que substitui o especificado no projeto de build.
- **`environmentTypeOverride`**: opcional. Um tipo de contêiner para esse build que substitui o especificado no projeto de build. A string atual válida é `LINUX_CONTAINER`.
- Os seguintes espaços reservados são para: `environmentVariablesOverride`.
 - **`environmentVariablesOverride-name`**: opcional. O nome de uma variável de ambiente no projeto de build, cujo valor você deseja sobrescrever para este build.
 - **`environmentVariablesOverride-type`**: opcional. O tipo de variável de ambiente no projeto de compilação cujo valor você deseja substituir para essa compilação.
 - **`environmentVariablesValue`**: opcional. O valor da variável de ambiente definida no projeto de compilação que você deseja substituir para essa compilação.
- **`gitCloneDepthOverride`**: opcional. O valor de Git clone depth no projeto de compilação que você deseja substituir para esta compilação. Se o tipo de origem é o Amazon S3, esse valor não é compatível.
- **`imageOverride`**: opcional. O nome de uma imagem para esse build que substitui o especificado no projeto de build.
- **`idempotencyToken`**: opcional. Uma string que serve como token para especificar que a solicitação de build é idempotente. Você pode escolher qualquer string com 64 caracteres ou menos. O token é válido por 5 minutos após a solicitação `start-build`. Se você repetir a solicitação `start-build` com o mesmo token, mas alterar um parâmetro, CodeBuild retornará um erro de incompatibilidade de parâmetros.
- **`insecureSslOverride`**: booleano opcional que especifica se a configuração TLS insegura especificada no projeto de construção deve ser substituída. A configuração de TLS insegura determina ignorar ou não avisos do TLS ao conectar-se ao código-fonte do projeto. Essa substituição se aplica somente se a origem da compilação for o GitHub Enterprise Server.
- **`privilegedModeOverride`**: booleano opcional. Se definido como verdadeiro, o build substitui o modo privilegiado no projeto de build.

- ***queuedTimeoutInMinutesOverride***: número inteiro opcional que especifica o número de minutos em que uma compilação pode ser colocada na fila antes de atingir o tempo limite. O valor mínimo é de cinco minutos e o valor máximo é 480 minutos (oito horas).
- ***reportBuildStatusOverride***: booleano opcional que especifica se você deve enviar ao provedor de origem o status de início e conclusão de uma compilação. Se você definir isso com um provedor de origem diferente do GitHub GitHub Enterprise Server ou do Bitbucket, um `invalidInputException` será lançado.
- ***sourceAuthOverride***: Cadeia de caracteres opcional. Um tipo de autorização para este build que sobrescreve a definida no projeto build. Essa substituição se aplica somente se a origem do projeto de compilação for Bitbucket ou. GitHub
- ***sourceLocationOverride***: Cadeia de caracteres opcional. Um local que sobrescreve neste build o local de origem definido no projeto de build.
- ***serviceRoleOverride***: Cadeia de caracteres opcional. O nome de uma função de serviço para esse build que substitui o especificado no projeto de build.
- ***sourceTypeOverride***: Cadeia de caracteres opcional. Um tipo de origem neste build que sobrescreve a origem definida no projeto de build. As strings válidas são `NO_SOURCE`, `CODECOMMIT`, `CODEPIPELINE`, `GITHUB`, `S3`, `BITBUCKET`, e `GITHUB_ENTERPRISE`.
- ***timeoutInMinutesOverride***: Número opcional. O número de minutos de tempo limite do build que sobrescreve para este build aquele definido no projeto de build.

Recomendamos que você armazene uma variável de ambiente com um valor confidencial, como um ID de chave de AWS acesso, uma chave de acesso AWS secreta ou uma senha como parâmetro no Amazon EC2 Systems Manager Parameter Store. CodeBuild pode usar um parâmetro armazenado no Amazon EC2 Systems Manager Parameter Store somente se o nome desse parâmetro começar com `/CodeBuild/` (por exemplo, `/CodeBuild/dockerLoginPassword`). Você pode usar o CodeBuild console para criar um parâmetro no Amazon EC2 Systems Manager. Escolha `Create a parameter` (Criar um parâmetro) e siga as instruções. (Nessa caixa de diálogo, para a chave KMS, você pode especificar opcionalmente o ARN de uma AWS KMS chave na sua conta. O Amazon EC2 Systems Manager usa essa chave para criptografar o valor do parâmetro durante o armazenamento e descriptografar durante a recuperação.) Se você usar o CodeBuild console para criar um parâmetro, o console iniciará o parâmetro `/CodeBuild/` conforme ele está sendo armazenado. No entanto, se você usar o console do Amazon EC2 Systems Manager Parameter Store para criar um parâmetro, deverá iniciar o nome do parâmetro com `/CodeBuild/` e definir `Type` como `Secure String`. Para obter mais informações, consulte [Armazenamento de AWS Systems Manager parâmetros](#) e

[explicação passo a passo: Criar e testar um parâmetro de string \(console\) no Guia](#) do usuário do Amazon EC2 Systems Manager.

Se seu projeto de construção se referir a parâmetros armazenados no Amazon EC2 Systems Manager Parameter Store, a função de serviço do projeto de construção deve permitir a `ssm:GetParameters` ação. Se você escolheu Criar uma nova função de serviço em sua conta anteriormente, CodeBuild inclua essa ação na função de serviço padrão do seu projeto de compilação automaticamente. No entanto, se escolher Choose an existing service role from your account, você deverá incluir essa ação na função de serviço separadamente.

As variáveis de ambiente definidas por você substituem variáveis de ambiente existentes. Por exemplo, se a imagem de Docker já contiver uma variável de ambiente chamada `MY_VAR` com um valor de `my_value` e você definir uma variável de ambiente chamada `MY_VAR` com um valor de `other_value`, `my_value` será substituído por `other_value`. Da mesma maneira, se a imagem de Docker já contiver uma variável de ambiente chamada `PATH` com um valor de `/usr/local/sbin:/usr/local/bin` e você definir uma variável de ambiente chamada `PATH` com um valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` será substituído pelo valor literal `$PATH:/usr/share/ant/bin`.

Não defina nenhuma variável de ambiente com um nome que comece com `CODEBUILD_`. Este prefixo está reservado para uso interno da .

Se uma variável de ambiente com o mesmo nome for definida em vários locais, o valor da variável de ambiente será determinado como se segue:

- O valor na chamada de operação de início de build tem a maior prioridade.
- O valor na definição de projeto de build tem a precedência seguinte.
- O valor na declaração do arquivo `buildspec` tem a menor prioridade.

Para obter informações sobre valores válidos para esses espaços reservados, consulte [Criar um projeto de compilação \(AWS CLI\)](#). Para obter uma lista das últimas configurações de um projeto de compilação, consulte [Exibir os detalhes do projeto de compilação](#).

5. Alterne para o diretório que contenha o arquivo que você acabou de salvar e execute o comando `start-build` novamente.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. Se bem-sucedidos, dados semelhantes aos descritos no procedimento [Para executar a compilação](#) serão exibidos na saída.

Para trabalhar com informações detalhadas sobre este build, anote o valor `id` da saída e consulte [Visualizar detalhes de compilação \(AWS CLI\)](#).

Executar uma compilação em lote (AWS CLI)

1. Execute o comando `start-build-batch` de uma das seguintes maneiras:

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

Use esta maneira se você quiser executar um build que utiliza a versão mais recente do artefato de entrada de build e as configurações existentes do projeto de build.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Use esta maneira se quiser executar uma compilação com uma versão anterior do artefato de entrada da compilação ou se quiser substituir as configurações dos artefatos de saída da compilação, as variáveis de ambiente, o `buildspec` ou o período de tempo limite padrão da compilação.

2. Se você executar o `start-build-batch` comando com a `--project-name` opção, `<project-name>` substitua pelo nome do projeto de compilação e, em seguida, vá para a etapa 6 desse procedimento. Para obter uma lista de projetos de compilação, consulte [Exibir os nomes do projeto de compilação](#).
3. Se você executar o comando `start-build-batch` com a opção `--idempotency-token`, um identificador, ou um token, exclusivo que diferencia maiúsculas e minúsculas será incluído na solicitação `start-build-batch`. O token é válido por 5 minutos após a solicitação. Se você repetir a `start-build-batch` solicitação com o mesmo token, mas alterar um parâmetro, CodeBuild retornará um erro de incompatibilidade de parâmetros.
4. Se você executar o `start-build-batch` comando com a `--generate-cli-skeleton` opção, os dados formatados em JSON serão enviados para o arquivo. `<json-file>` Esse arquivo é semelhante ao esqueleto produzido pelo comando `start-build`, com a adição do objeto a seguir. Para obter mais informações sobre os objetos comuns, consulte [Executar uma compilação \(AWS CLI\)](#).

Modifique esse arquivo para adicionar qualquer substituição de compilação e salve os resultados.

```
"buildBatchConfigOverride": {  
  "combineArtifacts": combineArtifacts,  
  "restrictions": {  
    "computeTypesAllowed": [  
      allowedComputeTypes  
    ],  
    "maximumBuildsAllowed": maximumBuildsAllowed  
  },  
  "serviceRole": "batchServiceRole",  
  "timeoutInMins": batchTimeout  
}
```

O `buildBatchConfigOverride` objeto é uma [ProjectBuildBatchConfig](#) estrutura que contém as substituições de configuração de compilação em lote para essa compilação.

combineArtifacts

Um valor booleano que especifica se os artefatos da compilação em lote devem ser combinados em um único local de artefato.

allowedComputeTypes

Uma matriz de strings que especificam os tipos de computação que são permitidos para a compilação em lote. Consulte [Build environment compute types](#) para obter esses valores.

maximumBuildsAllowed

Especifica o número máximo de criações permitidas.

batchServiceRole

Especifica o ARN da função de serviço para o projeto de compilação em lote.

batchTimeout

Especifica a quantidade máxima de tempo, em minutos, em que a compilação em lote deve ser concluída.

5. Alterne para o diretório que contenha o arquivo que você acabou de salvar e execute o comando `start-build-batch` novamente.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

- Se for bem-sucedida, a representação JSON de um [BuildBatch](#) objeto aparecerá na saída do console. Consulte a [sintaxe de StartBuildBatch resposta](#) para ver um exemplo desses dados.

Começar a executar compilações automaticamente (AWS CLI)

Se seu código-fonte estiver armazenado em um repositório GitHub ou em um repositório do GitHub Enterprise Server, você poderá usar GitHub webhooks para AWS CodeBuild reconstruir seu código-fonte sempre que uma alteração de código for enviada para o repositório.

Execute o comando create-webhook, como se segue:

```
aws codebuild create-webhook --project-name <project-name>
```

<project-name> é o nome do projeto de construção que contém o código-fonte a ser reconstruído.

Pois GitHub, informações semelhantes às seguintes aparecem na saída:

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

<url> é a URL do GitHub webhook.

Para o GitHub Enterprise Server, informações semelhantes às seguintes aparecem na saída:

```
{
  "webhook": {
    "secret": "YRV4JYAGFsekJiirp5ytx86oZpyhUdySNSDTLNUxoXX1c7aZ6XYDf37-ZFyY02rs4JSE70mLW3w-gh-ryoVB80SSSC1aAtBtuPkHwYuncCCmdogCVCfniQ7ukYX2_xM--nLDma5EngIg_Bi_N465yi33zyTUNPoQlxCpLO-BwghcVa9lAurwR77-uY7i-_XCJFahwMx1f4ubOgBBsMT2A16apqjQJoKSb61XVKyZy1Giuy4nliAXfv9WnN76CaCsndb3fVIE78fpygfo41xYxSQ6vpo6LRTKtPzbyeTHbVXGda1PJvnbK1nKmJDo0RTgI1m2oYr17dwziQlrrvoCoNgy1S00_7LKfA-nNXFc_f1SiFy0AqeMB43-d00cdkzybHncE81QTRwEUCFfmX-AJCwmlXV0kg0G67T92Sjbpz0fRlkh5pwIF193_bB_j0HDinK6i0iPpf2dIDAIZgGMagqZeWb-axDeTAbopoU8J6gFI1yKo5aq9q151zC1PERUsMgJFtJr_a-Z-L_kylr-4hSSxasSJNuJ43_X0BRWqT51xqvH-A69bv07KbVT_Kc6wxkSHyYCEMoa_Pfa7ZQgyfY6B00ogMNj31yFbjthORNL1cDo6-3J-McDLoyrRtSEOV9QnxvsG5zu1N5-z20rkJtg_MOfNwocfUutFXb7vrGTduH1R1dzXLRusHuxOVVuDUWm9vhwMr-hUkeGo_1kDKyk4E2QFvZxpjYw0vFfV-dwxFRr_mifzxWlwyfMt2iFtLkp_YZj_4WeFAckGefr-ilNaYvsZpzXj78Ae1adVolF48AmDdN2pWslWJjatU9zt942gLisFFmKakcvJuy5yxXHaxxbhUyC8NHYIESUWpfcfnqrMsR8op3P4AUCHIpiZCYyuiwI_cac-pIUB00Xaur_lu_fyFghgOJc7cfTnA36rv5X5DnFDM8P3HNBELjaF9QZ6AiJegPEWTHIKJON3AUDwpkz_hwTXyUoAU8MdZfPTXbBoT6N5Z5THBHsYxR",
    "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?t=eyJlbmNyeXB0ZWREYXRhIjoiUmFqMmJERGRQbGhwLzNTNld3R0VGRjZzOTNwLzZlZGV6NGZlIR1E0RUscxdzhGeWlnVFFfQWTR0WFwT2dJRnNmRHc3S3RNc0xYMEhNcXFTagk1cE1nSy9zPSIsIm12UGFyYW1ldGVyU3B1YyI6IndSQ1Qrc2VpQjBCZzhPeVYiLCJtYXR1cm1hbFNldFh1cm1hbCI6MX0%3D&v=1"
  }
}
```

- Copie a chave secreta e a URL da carga da saída. Você precisa que eles adicionem um webhook no GitHub Enterprise Server.

2. No GitHub Enterprise Server, escolha o repositório em que seu CodeBuild projeto está armazenado. Escolha Configurações, escolha Hooks e serviços, e depois escolha Adicionar Webhook.
3. Insira a URL da carga útil e a chave secreta, aceite os valores padrão para os outros campos e, em seguida, escolha Adicionar webhook.

Parar de executar compilações automaticamente (AWS CLI)

Se seu código-fonte estiver armazenado em um repositório GitHub ou em um repositório do GitHub Enterprise Server, você poderá configurar GitHub webhooks para AWS CodeBuild reconstruir seu código-fonte sempre que uma alteração de código for enviada para o repositório. Para obter mais informações, consulte [Começar a executar compilações automaticamente \(AWS CLI\)](#).

Se tiver habilitado esse comportamento, você poderá desativá-lo executando o comando `delete-webhook` da seguinte maneira:

```
aws codebuild delete-webhook --project-name <project-name>
```

- onde *<project-name>* é o nome do projeto de construção que contém o código-fonte a ser reconstruído.

Se esse comando for bem-sucedido, nenhuma informação e nenhum dado será exibido na saída.

Note

Isso exclui o webhook somente do seu CodeBuild projeto. Você também deve excluir o webhook do seu repositório GitHub ou do GitHub Enterprise Server.

Executar uma compilação (AWS SDKs)

Para usar CodePipeline para executar uma compilação com AWS CodeBuild, pule essas etapas e siga as instruções em [Use AWS CodeBuild with AWS CodePipeline para testar o código e executar compilações](#) vez disso.

Para obter informações sobre como usar CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Execute compilações em computação AWS Lambda

AWS Lambda a computação oferece velocidades de inicialização otimizadas para suas compilações. AWS Lambda suporta compilações mais rápidas devido a uma menor latência de inicialização. AWS Lambda também é dimensionado automaticamente, para que as compilações não fiquem esperando na fila para serem executadas. No entanto, existem alguns casos de uso que AWS Lambda não são compatíveis e, se afetarem você, use a EC2 computação. Para obter mais informações, consulte [Limitações da AWS Lambda computação](#).

Tópicos

- [Quais ferramentas e runtimes serão incluídos nas imagens do Docker do ambiente de runtime selecionado que são executadas no AWS Lambda?](#)
- [E se a imagem selecionada não incluir as ferramentas de que preciso?](#)
- [Em quais regiões oferecem suporte à AWS Lambda computação? CodeBuild](#)
- [Limitações da AWS Lambda computação](#)
- [Implemente uma função Lambda usando o AWS SAM CodeBuild Lambda Java](#)
- [Crie um aplicativo React de página única com CodeBuild Lambda Node.js](#)
- [Atualize a configuração de uma função Lambda com o CodeBuild Lambda Python](#)

Quais ferramentas e runtimes serão incluídos nas imagens do Docker do ambiente de runtime selecionado que são executadas no AWS Lambda?

AWS Lambda oferece suporte às seguintes ferramentas: AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby e .NET.

E se a imagem selecionada não incluir as ferramentas de que preciso?

Se a imagem selecionada não incluir as ferramentas necessárias, você pode fornecer uma imagem do Docker do ambiente personalizado que inclui as ferramentas necessárias.

Note

O Lambda não oferece suporte a funções que usam imagens de contêiner multiarquitetura. Para obter mais informações, consulte [Criar uma função Lambda usando uma imagem de contêiner](#) no Guia do AWS Lambda desenvolvedor.

Observe que você precisa das seguintes permissões do Amazon ECR para usar imagens personalizadas para computação Lambda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "arn:aws:ecr:image-region:image-account-id:repository/image-repo"
    }
  ]
}
```

Observe também que `curl` ou `wget` deve ser instalado para usar imagens personalizadas.

Em quais regiões oferecem suporte à AWS Lambda computação? CodeBuild

Em CodeBuild, a AWS Lambda computação é suportada no seguinte Regiões da AWS: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt), Europa (Irlanda) e América do Sul (São Paulo). Para obter mais informações sobre Regiões da AWS onde CodeBuild está disponível, consulte [AWS Serviços por região](#).

Limitações da AWS Lambda computação

Há alguns casos de uso que AWS Lambda não são compatíveis e, se afetarem você, use a EC2 computação:

- AWS Lambda não oferece suporte a ferramentas que exigem permissões de root. Para ferramentas como yum ou rpm, use o tipo de EC2 computação ou outras ferramentas que não exijam permissões de root.
- AWS Lambda não suporta compilações ou execuções do Docker.
- AWS Lambda não suporta gravação em arquivos externos/tmp. Os gerenciadores de pacotes incluídos estão configurados para usar o diretório /tmp por padrão para baixar e referenciar pacotes.
- AWS Lambda não oferece suporte ao tipo de ambiente LINUX_GPU_CONTAINER e não é compatível com o Windows Server Core 2019.
- AWS Lambda não suporta armazenamento em cache, tempos limite de compilação personalizados, tempo limite de fila, emblemas de construção, modo privilegiado, ambientes de tempo de execução personalizados ou tempos de execução superiores a 15 minutos.
- AWS Lambda não suporta conectividade VPC, um intervalo fixo de endereços IP de CodeBuild origem, EFS, instalação de certificados ou acesso SSH com o Session Manager.

Implemente uma função Lambda usando o AWS SAM CodeBuild Lambda Java

O AWS Serverless Application Model (AWS SAM) é uma estrutura de código aberto para criar aplicativos sem servidor. Para obter mais informações, consulte o [AWS Serverless Application Model repositório](#) em GitHub. O exemplo de Java a seguir usa o Gradle para criar e testar uma AWS Lambda função. Depois disso, a AWS SAM CLI é usada para implantar o AWS CloudFormation modelo e o pacote de implantação. Ao usar o CodeBuild Lambda, as etapas de construção, teste e implantação são todas gerenciadas automaticamente, permitindo que a infraestrutura seja atualizada rapidamente sem intervenção manual em uma única compilação.

Configurar seu AWS SAM repositório

Crie um AWS SAM Hello World projeto usando a AWS SAM CLI.

Para criar seu AWS SAM projeto

1. Siga as instruções no Guia do AWS Serverless Application Model desenvolvedor para [instalar a AWS SAM CLI](#) em sua máquina local.
2. Execute `sam init` e selecione a configuração de projeto a seguir.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions? [y/N]: N
Project name [sam-app]: <insert project name>
```

3. Carregue a pasta AWS SAM do projeto em um repositório de origem compatível. Para obter uma lista dos tipos de fonte compatíveis, consulte [ProjectSource](#).

Crie um projeto CodeBuild Lambda Java

Crie um projeto AWS CodeBuild Lambda Java e configure as permissões do IAM necessárias para a construção.

Para criar seu projeto CodeBuild Lambda Java

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.
4. Em Código-fonte, selecione o repositório de origem em que seu AWS SAM projeto está localizado.
5. Em Ambiente:
 - Para Computação, selecione Lambda.
 - Em Runtime(s), selecione Java.
 - Para Imagem, selecione `aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21`.

- Em Perfil de serviço, deixe a opção Novo perfil de serviço selecionada. Anote o nome da função. Isso será necessário quando você atualizar as permissões do IAM do projeto posteriormente neste exemplo.
6. Selecione Create build project (Criar projeto de compilação).
 7. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
 8. No painel de navegação, escolha Funções e selecione o perfil de serviço associado ao projeto. Você pode encontrar sua função de projeto em CodeBuild selecionando seu projeto de compilação, escolhendo Editar, Ambiente e, em seguida, Função de serviço.
 9. Escolha a guia Relacionamentos de confiança e, em seguida, escolha Editar política de confiança.
 10. Adicione a seguinte política em linha ao perfil do IAM. Isso será usado para implantar sua AWS SAM infraestrutura posteriormente. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "iam:*",
        "apigateway:*",
        "s3:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Configurar o buildspec do projeto

Para criar, testar e implantar sua função Lambda, CodeBuild lê e executa comandos de compilação a partir de um buildspec.

Para configurar o buildspec do projeto

1. No CodeBuild console, selecione seu projeto de compilação e escolha Editar e Buildspec.
2. Em Buildspec, escolha Inserir comandos de compilação e, em seguida, escolha Alternar para editor.
3. Exclua os comandos de compilação pré-preenchidos e cole o seguinte buildspec.

```
version: 0.2
env:
  variables:
    GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
      - sam build --template-file template.yaml
      - echo "Running deploy..."
      - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
      - yes | sam deploy
```

4. Selecione Atualizar buildspec.

Implemente sua infraestrutura AWS SAM Lambda

Use o CodeBuild Lambda para implantar automaticamente sua infraestrutura Lambda

Para implantar a infraestrutura do Lambda

1. Selecione Iniciar compilação. Isso criará, testará e implantará automaticamente seu AWS SAM aplicativo para AWS Lambda usando AWS CloudFormation.
2. Quando a compilação estiver concluída, navegue até o AWS Lambda console e pesquise sua nova função Lambda sob o nome do AWS SAM projeto.
3. Teste a função do Lambda selecionando API Gateway na visão geral da Função e clicando no URL do endpoint de API. Você deve ver uma página aberta com a mensagem "message": "hello world".

Limpar a infraestrutura

Para evitar cobranças adicionais pelos recursos que você usou durante este tutorial, exclua os recursos criados pelo seu AWS SAM modelo CodeBuild e.

Para limpar a infraestrutura

1. Navegue até o AWS CloudFormation console e selecione `aws-sam-cli-managed-default`.
2. Em Recursos, esvazie o bucket de implantação `SamCliSourceBucket`.
3. Exclua a pilha `aws-sam-cli-managed-default`.
4. Exclua a AWS CloudFormation pilha associada ao seu AWS SAM projeto. Essa pilha deve ter o mesmo nome do seu AWS SAM projeto.
5. Navegue até o CloudWatch console e exclua os grupos de CloudWatch registros associados ao seu CodeBuild projeto.
6. Navegue até o CodeBuild console e exclua seu CodeBuild projeto escolhendo Excluir projeto de compilação.

Crie um aplicativo React de página única com CodeBuild Lambda Node.js

[Create React App](#) é uma forma de criar aplicações React de página única. O exemplo de Node.js a seguir usa Node.js para compilar os artefatos de origem de Create React App e retorna os artefatos de compilação.

Configurar o repositório de origem e o bucket de artefatos

Crie um repositório de origem para o projeto usando yarn e Create React App.

Para configurar o repositório de origem e o bucket de artefatos

1. Na máquina local, execute `yarn create react-app <app-name>` para criar uma aplicação React simples.
2. Faça o upload da pasta do projeto da aplicação React em um repositório de origem compatível. Para obter uma lista dos tipos de fonte compatíveis, consulte [ProjectSource](#).

Crie um projeto CodeBuild Lambda Node.js

Crie um projeto AWS CodeBuild Lambda Node.js.

Para criar seu projeto CodeBuild Lambda Node.js

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.
4. Em Código-fonte, selecione o repositório de origem em que seu AWS SAM projeto está localizado.
5. Em Ambiente:
 - Para Computação, selecione Lambda.
 - Em Runtime(s), selecione Node.js.
 - Para Imagem, selecione aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.
6. Em Artefatos:
 - Em Tipo, selecione Amazon S3.
 - Em Nome do bucket, selecione o bucket de artefatos do projeto que você criou anteriormente.
 - Em Empacotamento de artefatos, selecione Zip.
7. Selecione Create build project (Criar projeto de compilação).

Configurar o buildspec do projeto

Para criar seu aplicativo React, CodeBuild lê e executa comandos de compilação a partir de um arquivo buildspec.

Para configurar o buildspec do projeto

1. No CodeBuild console, selecione seu projeto de compilação e escolha Editar e Buildspec.

2. Em Buildspec, escolha Inserir comandos de compilação e, em seguida, escolha Alternar para editor.
3. Exclua os comandos de compilação pré-preenchidos e cole o seguinte buildspec.

```
version: 0.2
phases:
  build:
    commands:
      - yarn
      - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
      - yarn run build
      - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
artifacts:
  name: "build-output"
  files:
    - "**/*"
reports:
  test-report:
    files:
      - 'junit.xml'
    file-format: 'JUNITXML'
  coverage-report:
    files:
      - 'coverage/clover.xml'
    file-format: 'CLOVERXML'
```

4. Selecione Atualizar buildspec.

Compilar e executar a aplicação React

Crie o aplicativo React no CodeBuild Lambda, baixe os artefatos de construção e execute o aplicativo React localmente.

Para compilar e executar a aplicação React

1. Selecione Iniciar compilação.
2. Quando a compilação estiver concluída, navegue até o bucket de artefatos do projeto Amazon S3 e baixe o artefato da aplicação React.

3. Descompacte o artefato de compilação do React e run `npm install -g serve && serve -s build` na pasta do projeto.
4. O comando `serve` servirá o site estático em uma porta local e imprimirá a saída no terminal. Você pode visitar o URL do localhost em Local: na saída do terminal para visualizar a aplicação React.

Para saber mais sobre como lidar com a implantação de um servidor baseado em React, consulte [Create React App Deployment](#).

Limpar a infraestrutura

Para evitar cobranças adicionais pelos recursos que você usou durante este tutorial, exclua os recursos criados para o seu CodeBuild projeto.

Para limpar a infraestrutura

1. Exclua o bucket do Amazon S3 dos artefatos do projeto
2. Navegue até o CloudWatch console e exclua os grupos de CloudWatch registros associados ao seu CodeBuild projeto.
3. Navegue até o CodeBuild console e exclua seu CodeBuild projeto escolhendo Excluir projeto de compilação.

Atualize a configuração de uma função Lambda com o CodeBuild Lambda Python

O exemplo de Python a seguir usa [Boto3](#) e CodeBuild Lambda Python para atualizar a configuração de uma função Lambda. Esse exemplo pode ser estendido para gerenciar outros AWS recursos de forma programática. Para obter mais informações, consulte a [documentação do Boto3](#).

Pré-requisitos

Crie ou encontre uma função do Lambda na sua conta.

Esse exemplo pressupõe que você já tenha criado uma função Lambda em sua conta e a CodeBuild usará para atualizar as variáveis de ambiente da função Lambda. Para obter mais informações sobre como configurar uma função Lambda por meio CodeBuild, consulte o [Implemente uma função Lambda usando o AWS SAM CodeBuild Lambda Java](#) exemplo ou visite. [AWS Lambda](#)

Configurar o repositório de origem

Crie um repositório de origem para armazenar o script Python do Boto3.

Para configurar o repositório de origem

1. Copie o script do Python a seguir em um novo arquivo chamado `update_lambda_environment_variables.py`.

```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
    print("Updating lambda function " + lambda_function_name + " environment
variable "
        + lambda_env_variable + " to " + lambda_env_variable_value)
    lambda_client.update_function_configuration(
        FunctionName=lambda_function_name,
        Environment={
            'Variables': {
                lambda_env_variable: lambda_env_variable_value
            }
        },
    )

if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Faça o upload do arquivo Python em um repositório de origem compatível. Para obter uma lista dos tipos de fonte compatíveis, consulte [ProjectSource](#).

Crie um projeto CodeBuild Lambda Python

Crie um projeto CodeBuild Lambda Python.

Para criar seu projeto CodeBuild Lambda Java

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se uma página de CodeBuild informações for exibida, escolha Criar projeto de construção. Caso contrário, no painel de navegação, expanda Compilar, escolha Projetos de compilação e, depois, Criar projeto de compilação.
3. Em Nome do projeto, insira um nome para esse projeto de compilação. Os nomes dos projetos de criação devem ser exclusivos em cada AWS conta. Também é possível incluir uma descrição opcional do projeto de compilação para ajudar outros usuários a entender para que esse projeto é usado.
4. Em Código-fonte, selecione o repositório de origem em que seu AWS SAM projeto está localizado.
5. Em Ambiente:
 - Para Computação, selecione Lambda.
 - Em Runtime(s), escolha Python.
 - Para Imagem, selecione aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12.
 - Em Perfil de serviço, deixe a opção Novo perfil de serviço selecionada. Anote o nome da função. Isso será necessário quando você atualizar as permissões do IAM do projeto posteriormente neste exemplo.
6. Selecione Create build project (Criar projeto de compilação).
7. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
8. No painel de navegação, escolha Funções e selecione o perfil de serviço associado ao projeto. Você pode encontrar sua função de projeto em CodeBuild selecionando seu projeto de compilação, escolhendo Editar, Ambiente e, em seguida, Função de serviço.
9. Escolha a guia Relacionamentos de confiança e, em seguida, escolha Editar política de confiança.
10. Adicione a seguinte política em linha ao perfil do IAM. Isso será usado para implantar sua AWS SAM infraestrutura posteriormente. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do IAM.

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

```
        "Effect": "Allow",
        "Action": [
            "lambda:UpdateFunctionConfiguration"
        ],
        "Resource": [
            "*"
        ]
    }
}
```

Configurar o buildspec do projeto

Para atualizar a função do Lambda, o script lê as variáveis de ambiente do buildspec para encontrar o nome da função do Lambda, o nome da variável de ambiente e o valor da variável de ambiente.

Para configurar o buildspec do projeto

1. No CodeBuild console, selecione seu projeto de compilação e escolha Editar e Buildspec.
2. Em Buildspec, escolha Inserir comandos de compilação e, em seguida, escolha Alternar para editor.
3. Exclua os comandos de compilação pré-preenchidos e cole o seguinte buildspec.

```
version: 0.2
env:
  variables:
    LAMBDA_FUNC_NAME: "<lambda-function-name>"
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
    LAMBDA_ENV_VARIABLE_VALUE: "true"
phases:
  install:
    commands:
      - pip3 install boto3
  build:
    commands:
      - python3 update_lambda_environment_variables.py
```

4. Selecione Atualizar buildspec.

Atualizar a configuração do Lambda

Use o CodeBuild Lambda Python para atualizar automaticamente a configuração da função Lambda.

Para atualizar a configuração da função do Lambda

1. Selecione Iniciar compilação.
2. Quando a compilação estiver concluída, navegue até a função do Lambda.
3. Escolha Configuração e, em seguida, as variáveis de Ambiente. Você deve ver uma nova variável de ambiente com a chave `FEATURE_ENABLED` e o valor `true`.

Limpar a infraestrutura

Para evitar cobranças adicionais pelos recursos que você usou durante este tutorial, exclua os recursos criados para o seu CodeBuild projeto.

Para limpar a infraestrutura

1. Navegue até o CloudWatch console e exclua os grupos de CloudWatch registros associados ao seu CodeBuild projeto.
2. Navegue até o CodeBuild console e exclua seu CodeBuild projeto escolhendo Excluir projeto de compilação.
3. Se você criou uma função do Lambda para o propósito deste exemplo, escolha Ações e Excluir função para limpar a função do Lambda.

Extensões

Se você quiser estender essa amostra para gerenciar outros AWS recursos usando o AWS CodeBuild Lambda Python:

- Atualize o script Python para modificar os novos recursos usando o Boto3.
- Atualize o papel do IAM associado ao seu CodeBuild projeto para ter permissões para os novos recursos.
- Adicione quaisquer novas variáveis de ambiente associadas aos novos recursos ao buildspec.

Executar compilações em frotas de capacidade reservada

CodeBuild oferece as seguintes frotas de computação:

- Frotas sob demanda
- Frotas de capacidade reservada

Com frotas sob demanda, CodeBuild fornece computação para suas construções. As máquinas são destruídas quando a compilação termina. As frotas sob demanda são totalmente gerenciadas e incluem recursos de escalabilidade automática para lidar com picos de demanda.

Note

As frotas sob demanda não são compatíveis com macOS.

CodeBuild também oferece frotas de capacidade reservada que contêm instâncias desenvolvidas pela Amazon EC2 que são mantidas pela CodeBuild. Com frotas de capacidade reservada, você configura um conjunto de instâncias dedicadas para seu ambiente de compilação. Essas máquinas permanecem ociosas, prontas para processar compilações ou testes imediatamente e reduzem a duração da compilação. Com frotas de capacidade reservada, suas máquinas estão sempre funcionando e continuarão a incorrer em custos enquanto forem provisionadas.

Important

Independentemente do tempo em que você executa uma instância, as frotas de capacidade reservada incorrem em uma cobrança inicial por instância, após a qual poderá haver custos associados adicionais. Para obter mais informações, consulte <https://aws.amazon.com/codebuild/pricing/>.

Tópicos

- [Criar uma frota de capacidade reservada](#)
- [Práticas recomendadas](#)
- [Posso compartilhar uma frota de capacidade reservada em vários CodeBuild projetos?](#)
- [Como funciona a computação baseada em atributos?](#)

- [Posso especificar manualmente uma EC2 instância da Amazon para minha frota?](#)
- [Quais regiões são compatíveis com frotas de capacidade reservada?](#)
- [Como configurar uma frota macOS de capacidade reservada?](#)
- [Como configuro uma Amazon Machine Image \(AMI\) personalizada para uma frota de capacidade reservada?](#)
- [Limitações de frotas de capacidade reservada](#)
- [Propriedades da frota de capacidade reservada](#)
- [Amostras de capacidade reservada com o AWS CodeBuild](#)

Criar uma frota de capacidade reservada

Use as instruções a seguir para criar uma frota de capacidade reservada.

Para criar uma frota de capacidade reservada

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Frotas de computação e Criar frota.
3. No campo de texto Nome da frota de computação, insira um nome para a frota.
4. No menu suspenso Sistema operacional, escolha o sistema operacional.
5. No menu suspenso Arquitetura, escolha a arquitetura.
6. (Opcional) Selecione Usar modo de execução da instância - opcional para execução direta em uma EC2 instância da Amazon em vez de em um contêiner Docker. Em seguida, escolha uma versão principal e uma versão secundária.
7. (Opcional) Em Configuração adicional, faça o seguinte:
 - Selecione Configurar VPC - opcional para conectar sua frota a uma VPC para acessar recursos privados durante o uso.
 - No menu suspenso VPC, selecione uma VPC que sua frota acessará. CodeBuild
 - No menu suspenso Sub-redes, selecione as sub-redes que CodeBuild devem ser usadas para definir sua configuração de VPC.
 - No menu suspenso Grupos de segurança, selecione os grupos de segurança que CodeBuild devem ser usados para trabalhar com sua VPC.
 - No campo Perfil de serviço de frota, escolha um perfil de serviço existente.

 Note

Verifique se a função da frota tem as permissões necessárias. Para obter mais informações, consulte [Permitir que um usuário adicione uma política de permissão para um perfil de serviço de frota](#).

- Se você escolheu o sistema operacional Amazon Linux, selecione Definir configurações de proxy - opcional para aplicar o controle de acesso à rede para as instâncias de capacidade reservada.
- Em Comportamento padrão, escolha permitir ou negar tráfego de saída para todos os destinos por padrão.
- Em Regras de proxy, escolha Adicionar regra de proxy para especificar domínios de destino ou IPs para permitir ou negar o controle de acesso à rede.
- Selecione Configurar AMI personalizada - opcional para usar uma Amazon Machine Image (AMI) personalizada.
 - No menu suspenso AMI, selecione uma Amazon Machine Image (AMI) para sua frota.
 - No campo Perfil de serviço de frota, escolha um perfil de serviço existente.

 Note

Verifique se a função da frota tem as permissões necessárias. Para obter mais informações, consulte [Permitir que um usuário adicione uma política de permissão para um perfil de serviço de frota](#).

8. Em Configuração de capacidade, no modo de seleção de computação, escolha uma das seguintes opções:

- Se você escolher Seleção guiada, faça o seguinte:
 - Para Computação, escolha o tipo de instâncias incluídas nessa frota.
 - No campo de texto Capacidade, insira o número mínimo de instâncias na frota.
 - (Opcional) Em Configuração adicional, faça o seguinte:
 - Selecione Configurar escalabilidade - opcional para escalar automaticamente sua frota com base nessa configuração. No menu suspenso Modo de escalabilidade - opcional, escolha o comportamento quando a demanda excede a capacidade da frota.
- Se você escolher Instância personalizada, faça o seguinte:

- No menu suspenso Tipo de instância de computação, selecione o tipo de instâncias incluídas nessa frota.
 - No campo de texto Tamanho adicional do volume do EBS - opcional, insira o volume adicional aos 64 GB de espaço em disco fornecidos.
 - No campo de texto Capacidade, insira o número mínimo de instâncias na frota.
 - (Opcional) Em Configuração adicional, faça o seguinte:
 - Selecione Configurar escalabilidade - opcional para escalar automaticamente sua frota com base nessa configuração. No menu suspenso Modo de escalabilidade - opcional, escolha o comportamento quando a demanda excede a capacidade da frota.
9. Escolha Criar frota de computação.
10. Depois que a frota computacional for criada, crie um novo CodeBuild projeto ou edite um existente. Em Ambiente, escolha Capacidade reservada em Modelo de provisionamento e escolha a frota especificada em Nome da frota.

Práticas recomendadas

Ao usar frotas de capacidade reservada, recomendamos que você siga estas práticas recomendadas.

- Recomendamos usar o modo de cache de origem para ajudar a melhorar o desempenho da compilação armazenando a fonte em cache.
- Recomendamos usar o cache de camadas do Docker para ajudar a melhorar o desempenho da compilação armazenando em cache as camadas existentes do Docker.

Posso compartilhar uma frota de capacidade reservada em vários CodeBuild projetos?

Sim, você pode maximizar a utilização da capacidade de uma frota usando-a em vários projetos.

Important

Ao usar o recurso de capacidade reservada, os dados armazenados em cache nas instâncias da frota, incluindo arquivos de origem, camadas do Docker e diretórios em cache especificados no buildspec, podem ser acessados por outros projetos na mesma conta. Isso

ocorre por design e permite que projetos dentro da mesma conta compartilhem instâncias de frota.

Como funciona a computação baseada em atributos?

Se você escolher `ATTRIBUTE_BASED_COMPUTE` como a sua `frota.computeType`, poderá especificar os atributos em um novo campo chamado `computeConfiguration`. Esses atributos incluem vCPUs, memória, espaço em disco e `machineType`. Isso `machineType` é um `GENERAL` ou outro `NVME`. Depois de especificar um ou alguns dos atributos disponíveis, CodeBuild escolherá um tipo de computação entre os tipos de instância compatíveis disponíveis como finalizado. `computeConfiguration`

Note

CodeBuild escolherá a instância mais barata que atenda a todos os requisitos de entrada. A memória CPUs, v e espaço em disco das instâncias escolhidas serão todos maiores ou iguais aos requisitos de entrada. Você pode verificar o resultado `computeConfiguration` na frota criada ou atualizada.

Se você inserir uma informação `computeConfiguration` que não seja possível de satisfazer CodeBuild, você receberá uma exceção de validação. Observe também que o comportamento de estouro de frota sob demanda será substituído pelo comportamento da fila se não estiver disponível sob demanda. `computeConfiguration`

Posso especificar manualmente uma EC2 instância da Amazon para minha frota?

Sim, você pode inserir diretamente a EC2 instância da Amazon desejada no console selecionando Instância personalizada ou configurando o parâmetro da API, `InstanceType`. Esse campo é usado no seguinte APIs: `CreateFleet` `UpdateFleet` `CreateProject`, `UpdateProject` `StartBuild` e. Para obter mais informações, consulte [Compute instance type](#).

Quais regiões são compatíveis com frotas de capacidade reservada?

Capacidade reservada As frotas Amazon Linux e Windows são suportadas no seguinte Regiões da AWS: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon), Ásia-

Pacífico (Mumbai), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt), Europa (Irlanda) e América do Sul (São Paulo). Para obter mais informações sobre Regiões da AWS onde CodeBuild está disponível, consulte [AWS Serviços por região](#).

As frotas de macOS Medium com capacidade reservada são suportadas no seguinte Regiões da AWS: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon), Ásia-Pacífico (Sydney) e Europa (Frankfurt). Capacidade reservada macOS Grandes frotas são compatíveis com o seguinte Regiões da AWS: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon) e Ásia-Pacífico (Sydney).

Como configurar uma frota macOS de capacidade reservada?

Para configurar uma frota macOS de capacidade reservada

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Frotas de computação e Criar frota.
3. No campo de texto Nome da frota de computação, insira um nome para a frota.
4. No menu suspenso Sistema operacional, escolha macOS.
5. No campo Computação, escolha um dos seguintes tipos de máquina de computação: Apple M2, 24 GB de memória, 8 v CPUs ou Apple M2, 32 GB de memória, 12 v. CPUs
6. No campo de texto Capacidade, insira o número mínimo de instâncias na frota.
7. (Opcional) Para usar uma imagem personalizada para sua frota, certifique-se de [Como configurar uma Amazon Machine Image \(AMI\) personalizada para uma frota de capacidade reservada?](#) que sua Amazon Machine Image (AMI) tenha os pré-requisitos necessários.
8. (Opcional) Para configurar uma VPC com a frota, em Configuração adicional, faça o seguinte:
 - No menu suspenso VPC - opcional, selecione uma VPC que sua frota acessará. CodeBuild
 - No menu suspenso Sub-redes, selecione as sub-redes que CodeBuild devem ser usadas para definir sua configuração de VPC.
 - No menu suspenso Grupos de segurança, selecione os grupos de segurança que CodeBuild devem ser usados para trabalhar com sua VPC.
 - No campo Perfil de serviço de frota, escolha um perfil de serviço existente.

 Note

Verifique se a função da frota tem as permissões necessárias. Para obter mais informações, consulte [Permitir que um usuário adicione uma política de permissão para um perfil de serviço de frota](#).

9. Escolha Criar frota de computação e aguarde a inicialização da instância da frota. Uma vez lançada, a capacidade será n/n , onde n está a capacidade fornecida.
10. Depois que a frota de computação for lançada, crie um novo CodeBuild projeto ou edite um existente. Em Ambiente, escolha Capacidade reservada em Modelo de provisionamento e escolha a frota especificada em Nome da frota.

Como configuro uma Amazon Machine Image (AMI) personalizada para uma frota de capacidade reservada?

Para configurar uma Amazon Machine Image (AMI) personalizada para uma frota de capacidade reservada

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Frotas de computação e Criar frota.
3. No campo de texto Nome da frota de computação, insira um nome para a frota.
4. Escolha uma imagem personalizada para sua frota e garanta que sua Amazon Machine Image (AMI) tenha os seguintes pré-requisitos:
 - Se seu tipo de ambiente for `MAC_ARM`, certifique-se de que sua arquitetura de AMI seja de 64 bits `Mac-Arm`.
 - Se seu tipo de ambiente for `LINUX_EC2`, certifique-se de que sua arquitetura de AMI seja de 64 bits `x86`.
 - Se seu tipo de ambiente for `ARM_EC2`, certifique-se de que sua arquitetura de AMI seja de 64 bits `Arm`.
 - Se seu tipo de ambiente for `WINDOWS_EC2`, certifique-se de que sua arquitetura de AMI seja de 64 bits `x86`.

- A AMI permite o ARN da organização de CodeBuild serviços. Para obter uma lista de organizações ARNs, consulte [Amazon Machine Images \(AMI\)](#).
- Se a AMI for criptografada com uma AWS KMS chave, a AWS KMS chave também deverá permitir a ID da organização do CodeBuild serviço. Para obter uma lista de organizações IDs, consulte [Amazon Machine Images \(AMI\)](#). Para obter mais informações sobre AWS KMS chaves, consulte [Permitir organizações e OUs usar uma chave KMS](#) no Guia do EC2 usuário da Amazon. Para dar permissão à CodeBuild organização para usar uma chave KMS, adicione a seguinte declaração à política de chaves:

```
{
  "Sid": "Allow access for organization root",
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
    "kms:Describe*",
    "kms:List*",
    "kms:Get*",
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalOrgID": "o-123example"
    }
  }
}
```

- No campo Função de serviço da frota, conceda as seguintes EC2 permissões da Amazon:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots"
      ]
    }
  ]
}
```

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

Limitações de frotas de capacidade reservada

Há alguns casos de uso que as frotas de capacidade reservada não permitem e, se afetarem você, use as frotas sob demanda:

- As frotas de capacidade reservada não oferecem suporte a métricas de utilização de compilações.
- As frotas macOS de capacidade reservada não são compatíveis com a sessão de depuração.

Consulte mais informações sobre cotas e limites em [Frotas de computação](#).

Propriedades da frota de capacidade reservada

Uma frota de capacidade reservada contém as seguintes propriedades. Para obter mais informações sobre frotas de capacidade reservada, consulte [Executar compilações em frotas de capacidade reservada](#).

Sistema operacional

O sistema operacional Os seguintes sistemas operacionais estão disponíveis:

- Amazon Linux
- macOS
- Windows Server 2019
- Windows Server 2022

Arquitetura

A arquitetura do processador. As seguintes arquiteturas estão disponíveis:

- x86_64
- Arm64

Tipo de ambiente

Os tipos de ambiente disponíveis quando o Amazon Linux é selecionado. Os seguintes tipos de ambiente estão disponíveis:

- Linux EC2
- GPU Linux

Tipo de instância de computação

As configurações de computação para instâncias de frota.

Seleção guiada

Especifique diferentes tipos de computação selecionando as configurações de vCPU, memória e espaço em disco. Para obter informações sobre a disponibilidade do tipo de computação por região, consulte [Sobre os tipos de ambiente de capacidade reservada](#).

Instância personalizada

Especifique manualmente o tipo de instância desejado.

Capacidade

O número inicial de máquinas alocadas para a frota, que define o número de compilações que podem ser executadas paralelamente.

Comportamento de estouro

Define o comportamento quando o número de compilações excede a capacidade da frota.

Sob demanda

As compilações do Overflow são executadas sob demanda. CodeBuild

Note

Se você optar por definir o comportamento de estouro como sob demanda ao criar uma frota conectada à VPC, adicione as permissões da VPC necessárias ao perfil de serviço do projeto. Para obter mais informações, consulte [Exemplo de declaração de política para permitir o CodeBuild acesso aos AWS serviços necessários para criar uma interface de rede VPC](#).

⚠ Important

Se você optar por definir seu comportamento de estouro como sob demanda, observe que as compilações de estouro serão cobradas separadamente, de forma semelhante à Amazon sob demanda. EC2 Para obter mais informações, consulte <https://aws.amazon.com/codebuild/pricing/>.

Queue (Fila)

As execuções de compilação são colocadas em uma fila até que uma máquina esteja disponível. Isso limita os custos adicionais porque nenhuma máquina adicional é alocada.

Imagens de máquina da Amazon (AMI)

As propriedades da imagem de máquina da Amazon (AMI) para sua frota. As seguintes propriedades são suportadas por CodeBuild:

Regiões da AWS	ARN de organização	ID da organização
us-east-1	arn:aws:organizations::851725618577:organization/o-c6wcu152r1	o-c6wcu152r1
us-east-2	arn:aws:organizations::992382780434:organization/o-seufr2suvq	o-seufr2suvq
us-west-2	arn:aws:organizations::381491982620:organization/o-0412o99a4r	o-0412o99a4r
ap-northeast-1	arn:aws:organizations::891376993293:organization/o-b6k3sjqavm	o-b6k3sjqavm

Regiões da AWS	ARN de organização	ID da organização
ap-south-1	arn:aws:organizations::891376924779:organization/o-krtah1lkeg	o-krtah1lkeg
ap-southeast-1	arn:aws:organizations::654654522137:organization/o-mcn8uvc3tp	o-mcn8uvc3tp
ap-southeast-2	arn:aws:organizations::767398067170:organization/o-6crt0f6bu4	o-6crt0f6bu4
eu-central-1	arn:aws:organizations::590183817084:organization/o-lb2lne3te6	o-lb2lne3te6
eu-west-1	arn:aws:organizations::891376938588:organization/o-ullrrg5qf0	o-ullrrg5qf0
sa-east-1	arn:aws:organizations::533267309133:organization/o-db63c45ozw	o-db63c45ozw

Configuração adicional

VPC - opcional

O VPC que sua CodeBuild frota acessará. Para obter mais informações, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

 Note

Se uma substituição de frota for especificada ao chamar a StartBuild API, a configuração da VPC do projeto CodeBuild será ignorada.

Sub-redes

As sub-redes de VPC usadas para definir sua configuração de VPC. CodeBuild Observe que as frotas de capacidade reservada oferecem suporte a apenas uma sub-rede em uma única zona de disponibilidade. Além disso, verifique se as sub-redes incluem um gateway NAT.

Grupos de segurança

Os grupos de segurança da VPC que são CodeBuild usados com sua VPC. Verifique se os grupos de segurança permitem conexões de saída.

Perfil de serviço de frota

Define o perfil de serviço para sua frota a partir de um perfil de serviço existente na conta.

Definir configurações de proxy - opcional

Configurações de proxy que aplicam controle de acesso à rede às instâncias de capacidade reservada. Para obter mais informações, consulte [Use AWS CodeBuild com um servidor proxy gerenciado](#).

 Note

As configurações de proxy não são compatíveis com VPC, Windows ou macOS.

Comportamento padrão

Define o comportamento do tráfego de saída.

Permitir

Permite tráfego de saída para todos os destinos por padrão.

Negar

Nega tráfego de saída para todos os destinos por padrão.

Regras de proxy

Especifica os domínios de destino aos quais IPs permitir ou negar o controle de acesso à rede.

Amostras de capacidade reservada com o AWS CodeBuild

Essas amostras podem ser usadas para fazer experiências com frotas de capacidade reservada em CodeBuild.

Tópicos

- [Armazenamento em cache com amostra de capacidade reservada](#)

Armazenamento em cache com amostra de capacidade reservada

Um cache pode armazenar partes reutilizáveis do seu ambiente de build e usá-las em vários builds. Este exemplo demonstrou como habilitar o armazenamento em cache no projeto de compilação usando a capacidade reservada. Para obter mais informações, consulte [Compilações em cache para melhorar o desempenho](#).

Você pode começar especificando um ou mais modos de cache nas configurações do projeto:

Cache:

Type: LOCAL

Modes:

- LOCAL_CUSTOM_CACHE
- LOCAL_DOCKER_LAYER_CACHE
- LOCAL_SOURCE_CACHE

Note

Habilite o modo privilegiado para usar o cache de camadas do Docker.

As configurações de especificação da compilação do projeto devem se parecer com o seguinte:

```
version: 0.2
phases:
```

```
build:
  commands:
    - echo testing local source cache
    - touch /codebuild/cache/workspace/foobar.txt
    - git checkout -b cached_branch
    - echo testing local docker layer cache
    - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
    - echo testing local custom cache
    - touch foo
    - mkdir bar && ln -s foo bar/foo2
    - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
    - "[ -f foo ] || exit 1"
    - "[ -L bar/foo2 ] || exit 1"
    - "[ -f bar/bar/foo3 ] || exit 1"
    - "[ -f bar/bar/foo4 ] || exit 1"
  cache:
    paths:
      - './foo'
      - './bar/**/*'
      - './bar/bar/foo3'
```

Você pode começar executando uma compilação com o novo projeto para propagar o cache. Ao concluir, você deve iniciar outra compilação com uma especificação de compilação de substituição, semelhante à seguinte:

```
version: 0.2
phases:
  build:
    commands:
      - echo testing local source cache
      - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
      - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
      - echo testing local docker layer cache
      - docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
      - echo testing local custom cache
      - "[ -f foo ] || exit 1"
      - "[ -L bar/foo2 ] || exit 1"
      - "[ -f bar/bar/foo3 ] || exit 1"
      - "[ -f bar/bar/foo4 ] || exit 1"
    cache:
      paths:
```

```
- './foo'
- './bar/**/*'
- './bar/bar/foo3'
```

Executar compilações em lotes

Você pode usar AWS CodeBuild para executar compilações simultâneas e coordenadas de um projeto com compilações em lote.

Tópicos

- [Perfil de segurança](#)
- [Tipos de compilação em lote](#)
- [Modo de relatório em lote](#)
- [Mais informações](#)

Perfil de segurança

As compilações em lote introduzem um novo perfil de segurança na configuração em lote. Essa nova função é necessária, pois CodeBuild você deve poder chamar as `RetryBuild` ações `StartBuild` `StopBuild`, e em seu nome para executar compilações como parte de um lote. Os clientes devem usar um novo perfil, e não o mesmo perfil que usam na compilação, por dois motivos:

- Fornecer ao perfil de compilação as permissões `StartBuild`, `StopBuild` e `RetryBuild` que permitem a uma única compilação iniciar mais compilações por meio do `buildspec`.
- CodeBuild compilações em lote fornecem restrições que restringem o número de compilações e tipos de computação que podem ser usados para as compilações no lote. Se o perfil de compilação tiver essas permissões, será possível que as próprias compilações ignorem essas restrições.

Tipos de compilação em lote

CodeBuild suporta os seguintes tipos de compilação em lote:

Tipos de compilação em lote

- [Grafo de compilação](#)

- [Lista de compilações](#)
- [Matriz de compilações](#)
- [Crie um fanout](#)

Grafo de compilação

Um grafo de compilação define um conjunto de tarefas que dependem de outras tarefas no lote.

O exemplo a seguir define um grafo de compilação que cria uma cadeia de dependências.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

Neste exemplo:

- A build1 é executada primeiro porque não tem dependências.

- A build2 tem uma dependência em build1, então a build2 é executada após a conclusão da build1.
- A build3 tem uma dependência em build2, então a build3 é executada após a conclusão da build2.

Para obter mais informações sobre a sintaxe buildspec do grafo de compilação, consulte [batch/build-graph](#).

Lista de compilações

Uma lista de compilações define várias tarefas que são executadas paralelamente.

O exemplo a seguir define uma lista de compilações. As compilações build1 e build2 serão executadas em paralelo.

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

Para obter mais informações sobre a sintaxe buildspec da lista de compilações, consulte [batch/build-list](#).

Matriz de compilações

Uma matriz de construção define tarefas com configurações diferentes que são executadas paralelamente. CodeBuild cria uma compilação separada para cada combinação de configuração possível.

O exemplo a seguir mostra uma matriz de compilações com dois arquivos buildspec e três valores para uma variável de ambiente.

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Neste exemplo, CodeBuild cria seis compilações:

- matrix1.yml com \$MY_VAR=VALUE1
- matrix1.yml com \$MY_VAR=VALUE2
- matrix1.yml com \$MY_VAR=VALUE3
- matrix2.yml com \$MY_VAR=VALUE1
- matrix2.yml com \$MY_VAR=VALUE2
- matrix2.yml com \$MY_VAR=VALUE3

Cada compilação terá as seguintes configurações:

- ignore-failure definido como false
- env/type definido como LINUX_CONTAINER

- `env/image` definido como `aws/codebuild/amazonlinux-x86_64-standard:4.0`
- `env/privileged-mode` definido como `true`

Essas compilações são executadas em paralelo.

Para obter mais informações sobre a sintaxe `buildspec` da matriz de compilações, consulte [batch/build-matrix](#).

Crie um fanout

Um fanout de compilação define uma tarefa que será dividida em várias compilações no lote. Isso pode ser usado para executar testes em paralelo. CodeBuild cria uma compilação separada para cada fragmento de casos de teste com base no valor definido no `parallelism` campo.

O exemplo a seguir define um fanout de compilação que cria cinco compilações que são executadas em paralelo.

```
version: 0.2

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

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

Neste exemplo, supondo que haja 100 testes que precisam ser executados, CodeBuild cria cinco compilações, cada uma executando 20 testes em paralelo.

Para obter mais informações sobre a sintaxe buildspec do grafo de compilação, consulte [batch/build-fanout](#).

Modo de relatório em lote

Se o provedor de origem do seu projeto for Bitbucket ou GitHub Enterprise e seu projeto estiver configurado para relatar os status de criação ao provedor de origem, você poderá selecionar como deseja que os status de criação em lote sejam enviados ao provedor de origem. GitHub É possível optar por enviar os status como um único relatório de status agregado para o lote ou ter o status de cada compilação no lote informado individualmente.

Para obter mais informações, consulte os tópicos a seguir.

- [Configuração em lote \(criar\)](#)
- [Configuração em lote \(atualizar\)](#)

Mais informações

Para obter mais informações, consulte os tópicos a seguir.

- [Referência de buildspec de compilação em lote](#)
- [Configuração em lote](#)
- [Executar uma compilação em lote \(AWS CLI\)](#)
- [Interrompa a criação de lotes AWS CodeBuild](#)

Execute testes paralelos em compilações em lote

Você pode usar AWS CodeBuild para executar testes paralelos em compilações em lote. A execução paralela de testes é uma abordagem de teste em que vários casos de teste são executados simultaneamente em diferentes ambientes, máquinas ou navegadores, em vez de serem executados sequencialmente. Essa abordagem pode reduzir significativamente o tempo geral de execução do teste e melhorar a eficiência do teste. Em CodeBuild, você pode dividir seus testes em vários ambientes e executá-los simultaneamente.

As principais vantagens da execução paralela de testes incluem:

1. Tempo de execução reduzido — testes que levariam horas sequencialmente podem ser concluídos em minutos.

2. Melhor utilização dos recursos - faz uso eficiente dos recursos de computação disponíveis.
3. Feedback antecipado - A conclusão mais rápida do teste significa um feedback mais rápido para os desenvolvedores.
4. Econômico - economiza tempo e custos de computação a longo prazo.

Ao implementar a execução paralela de testes, duas abordagens principais são comumente consideradas: ambientes separados e multithreading. Embora ambos os métodos tenham como objetivo alcançar a execução simultânea de testes, eles diferem significativamente em sua implementação e eficácia. Ambientes separados criam instâncias isoladas em que cada suíte de testes é executada de forma independente, enquanto o multithreading executa vários testes simultaneamente no mesmo espaço de processo usando diferentes threads.

As principais vantagens de ambientes separados em relação ao multithreading incluem:

1. Isolamento - Cada teste é executado em um ambiente completamente isolado, evitando interferência entre os testes.
2. Conflitos de recursos - Não há competição por recursos compartilhados, o que geralmente ocorre em multithreading.
3. Estabilidade - Menos propenso a condições de corrida e problemas de sincronização.
4. Depuração mais fácil - Quando os testes falham, é mais simples identificar a causa, pois cada ambiente é independente.
5. Gerenciamento de estado - gerencie facilmente os problemas de estado compartilhado que afetam os testes multiencaadeados.
6. Melhor escalabilidade - Pode adicionar facilmente mais ambientes sem complexidade.

Tópicos

- [Support em AWS CodeBuild](#)
- [Habilite a execução paralela de testes em compilações em lote](#)
- [Use o comando codebuild-tests-run CLI](#)
- [Use o comando codebuild-glob-search CLI](#)
- [Sobre a divisão de testes](#)
- [Mescle automaticamente relatórios de criação individuais](#)
- [Exemplo de execução paralela de testes para várias estruturas de teste](#)

Support em AWS CodeBuild

AWS CodeBuild fornece suporte robusto para execução paralela de testes por meio de seu recurso de criação em lote, projetado especificamente para aproveitar a execução de ambientes separados. Essa implementação se alinha perfeitamente aos benefícios de ambientes de teste isolados.

Compilação em lote com distribuição de teste

CodeBuildA funcionalidade de criação em lote do permite a criação de vários ambientes de construção que são executados simultaneamente. Cada ambiente opera como uma unidade completamente isolada, com seus próprios recursos computacionais, ambiente de execução e dependências. Por meio da configuração de compilação em lote, você pode especificar quantos ambientes paralelos eles precisam e como os testes devem ser distribuídos entre eles.

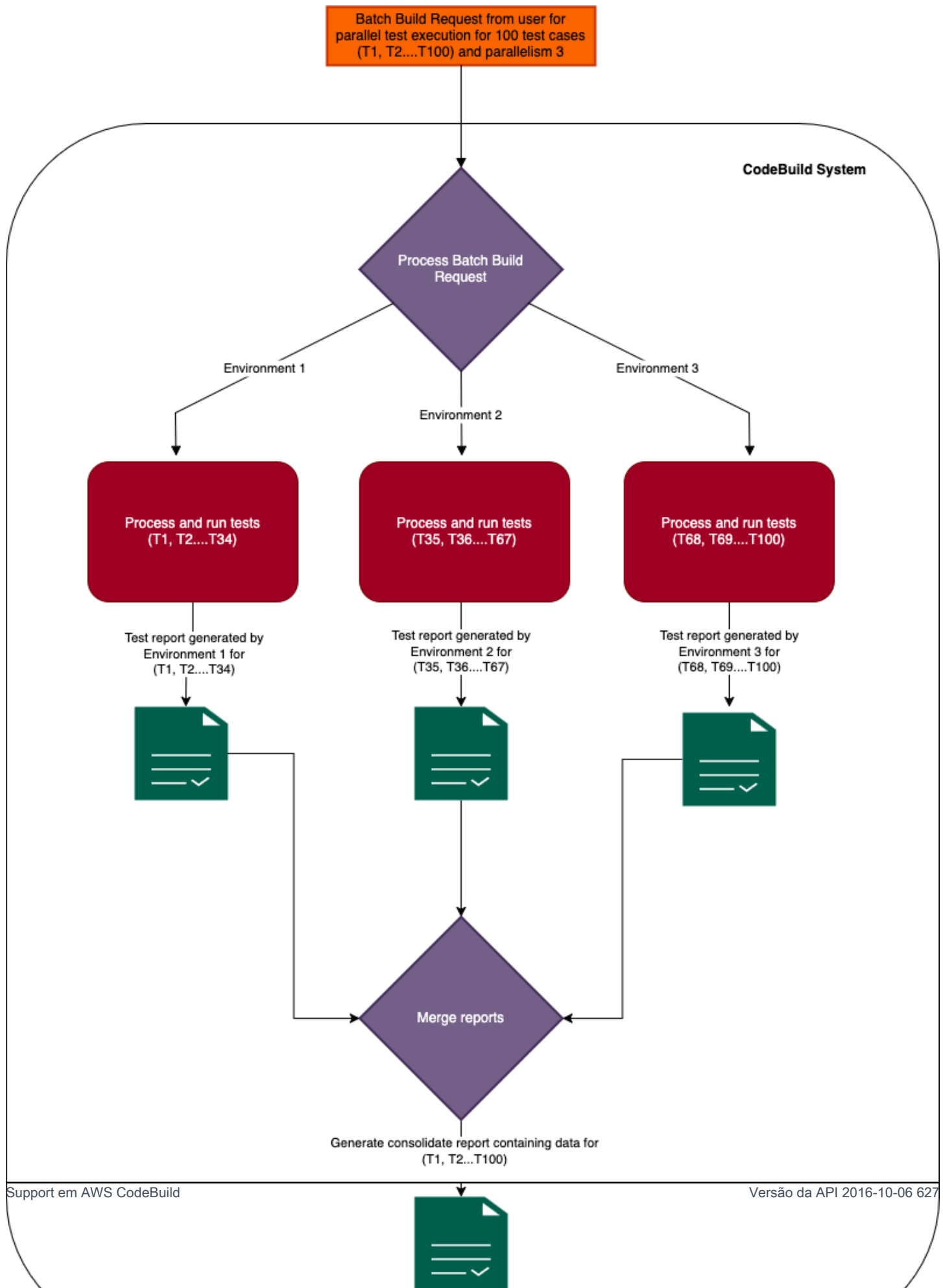
Teste a CLI de fragmentação

CodeBuild inclui um mecanismo de distribuição de testes integrado por meio de sua ferramenta CLI `codebuild-tests-run`, que divide automaticamente os testes em diferentes ambientes.

Agregação de relatórios

Um dos principais pontos fortes da implementação é a capacidade CodeBuild de lidar com a agregação de resultados de testes sem problemas. Enquanto os testes são executados em ambientes separados, coleta e combina CodeBuild automaticamente os relatórios de teste de cada ambiente em um relatório de teste unificado no nível de criação em lote. Essa consolidação fornece uma visão abrangente dos resultados dos testes, mantendo os benefícios de eficiência da execução paralela.

O diagrama a seguir explica o conceito completo de execução paralela de testes em AWS CodeBuild.



Habilite a execução paralela de testes em compilações em lote

Para executar testes em paralelo, atualize o arquivo `buildspec` de compilação em lote para incluir o campo `build-fanout` e o número de compilações paralelas para dividir o conjunto de testes no campo, conforme mostrado abaixo. `parallelism` O `parallelism` campo especifica quantos executores independentes estão configurados para executar a suíte de testes.

Para executar os testes em vários ambientes de execução paralela, defina o `parallelism` campo com um valor maior que zero. No exemplo abaixo, `parallelism` está definido como cinco, o que significa que CodeBuild inicia cinco compilações idênticas que executam uma parte da suíte de testes em paralelo.

Você pode usar o comando [codebuild-tests-run](#) CLI para dividir e executar seus testes. Seus arquivos de teste serão divididos e uma parte dos testes será executada em cada compilação. Isso reduz o tempo total necessário para executar a suíte de testes completa. No exemplo a seguir, os testes serão divididos em cinco e os pontos de divisão serão calculados com base no nome dos testes.

```
version: 0.2

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

phases:
  install:
    commands:
      - npm install jest-junit --save-dev
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/_tests_/**/*test.js'" \
          --sharding-strategy 'equal-distribution'

  post_build:
    commands:
```

```
- codebuild-glob-search '**/*.xml'
- echo "Running post-build steps..."
- echo "Build completed on `date`"

reports:
  test-reports:
    files:
      - '**/junit.xml'
    base-directory: .
    discard-paths: yes
    file-format: JUNITXML
```

Se os relatórios forem configurados para compilação build-fanout, os relatórios de teste serão gerados para cada compilação separadamente, que podem ser visualizados na guia Relatórios das compilações correspondentes no console. AWS CodeBuild

Para obter mais informações sobre como executar testes paralelos em lote, consulte [Exemplo de execução paralela de testes para várias estruturas de teste](#).

Use o comando **codebuild-tests-run** CLI

AWS CodeBuild fornece uma CLI que tomará o comando de teste e a localização do arquivo de teste como entrada. A CLI com essas entradas dividirá os testes em número de fragmentos, conforme especificado no `parallelism` campo, com base nos nomes dos arquivos de teste. A atribuição dos arquivos de teste ao fragmento é decidida pela estratégia de fragmentação.

```
codebuild-tests-run \
  --files-search "codebuild-glob-search '**/__tests__/*.js'" \
  --test-command 'npx jest --runInBand --coverage' \
  --sharding-strategy 'equal-distribution'
```

A tabela a seguir descreve os campos do comando `codebuild-tests-run` CLI.

Nome do campo	Tipo	Obrigatório ou opcional	Definição
<code>test-command</code>	String	Obrigatório	Esse comando é usado para executar os testes.

Nome do campo	Tipo	Obrigatório ou opcional	Definição
<code>files-search</code>	String	Obrigatório	Esse comando fornece uma lista de arquivos de teste. Você pode usar o comando codebuild -glob-search CLI AWS CodeBuild fornecido ou qualquer outra ferramenta de pesquisa de arquivos de sua escolha.  Note Certifique-se de que o <code>files-search</code> comando produza os nomes dos arquivos, cada um separado por uma nova linha.

Nome do campo	Tipo	Obrigatório ou opcional	Definição
sharding-strategy	Enum	Opcional	Valores válidos: <code>equal-distribution</code> (padrão), <code>stability</code> • <code>equal-distribution</code> : fragmente arquivos de teste uniformemente com base nos nomes dos arquivos de teste. • <code>stability</code> : fragmente arquivos de teste usando hash consistente dos nomes dos arquivos. Para obter mais informações, consulte Sobre a divisão de testes.

A `codebuild-tests-run` CLI funciona primeiro para identificar a lista de arquivos de teste usando o comando fornecido no `files-search` parâmetro. Em seguida, ele determina um subconjunto de arquivos de teste designados para o fragmento (ambiente) atual usando a estratégia de fragmentação especificada. Finalmente, esse subconjunto de arquivos de teste é formatado em uma lista separada por espaços e anexado ao final do comando fornecido no parâmetro antes de ser executado. `test-command`

Para estruturas de teste que não aceitam listas separadas por espaço, a `codebuild-tests-run` CLI fornece uma alternativa flexível por meio da variável de ambiente.

CODEBUILD_CURRENT_SHARD_FILES Essa variável contém uma lista separada por nova linha de caminhos de arquivo de teste designados para o fragmento de compilação atual. Ao aproveitar essa variável de ambiente, você pode se adaptar facilmente a vários requisitos da estrutura de teste, acomodando aqueles que esperam formatos de entrada diferentes das listas separadas por espaço. Além disso, você também pode formatar os nomes dos arquivos de teste conforme a necessidade da estrutura de teste. A seguir está um exemplo do uso de **CODEBUILD_CURRENT_SHARD_FILES** no Linux com a estrutura Django. Aqui **CODEBUILD_CURRENT_SHARD_FILES** é usado para obter caminhos de arquivo de notação de pontos suportados pelo Django:

```
codebuild-tests-run \
  -files-search "codebuild-glob-search '/tests/test_.py'" \
  -test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
-E "s/\//_/g; s/\.py$/_/; s/_/./g")' \
  -sharding-strategy 'equal-distribution'
```

Note

Observe que a variável de **CODEBUILD_CURRENT_SHARD_FILES** ambiente só pode ser usada dentro do escopo da `codebuild-tests-run` CLI.

Além disso, se você estiver usando o comando de teste **CODEBUILD_CURRENT_SHARD_FILES** interno, coloque **CODEBUILD_CURRENT_SHARD_FILES** entre aspas duplas, conforme mostrado no exemplo acima.

Use o comando **codebuild-glob-search** CLI

AWS CodeBuild fornece uma ferramenta CLI integrada chamada `codebuild-glob-search` que permite pesquisar arquivos em seu diretório de trabalho com base em um ou mais padrões globais. Essa ferramenta pode ser particularmente útil quando você deseja executar testes em arquivos ou diretórios específicos do seu projeto.

Uso

A `codebuild-glob-search` CLI tem a seguinte sintaxe de uso:

```
codebuild-glob-search <glob_pattern1> [<glob_pattern2> ...]
```

- *<glob_pattern1>*, *<glob_pattern2>*, etc.: um ou mais padrões globais para comparar com os arquivos em seu diretório de trabalho.
- *: corresponde a qualquer sequência de caracteres (excluindo separadores de caminho).
- **: corresponde a qualquer sequência de caracteres (incluindo separadores de caminho).

Note

Certifique-se de que a string global tenha aspas. Para verificar os resultados da correspondência de padrões, use o echo comando.

```
version: 0.2

phases:
  build:
    commands:
      - echo $(codebuild-glob-search '**/__tests__/*.js')
      - codebuild-glob-search '**/__tests__/*.js' | xargs -n 1 echo
```

Saída

A CLI produzirá uma lista separada por nova linha de caminhos de arquivo que correspondem aos padrões globais fornecidos. Os caminhos do arquivo retornados serão relativos ao diretório de trabalho.

Se nenhum arquivo for encontrado que corresponda aos padrões fornecidos, a CLI exibirá uma mensagem indicando que nenhum arquivo foi encontrado.

Observe que os diretórios encontrados devido a qualquer padrão serão excluídos dos resultados da pesquisa.

Exemplo

Se você quiser pesquisar somente arquivos dentro do diretório tests e seus subdiretórios com uma .js extensão, você pode usar o seguinte comando com a CLI `codebuild-glob-search`:

```
codebuild-glob-search '**/__tests__/*.js'
```

Esse comando procurará todos os arquivos com uma `.js` extensão dentro do `__tests__` diretório e seus subdiretórios, conforme indicado pelo padrão.

Sobre a divisão de testes

AWS CodeBuild O recurso de divisão de testes permite paralelizar a execução da suíte de testes em várias instâncias de computação, reduzindo o tempo geral de execução do teste. Esse recurso é ativado por meio da configuração em lote nas configurações CodeBuild do projeto e do `codebuild-tests-run` utilitário no arquivo `buildspec`.

Os testes são divididos com base na estratégia de fragmentação especificada. CodeBuild fornece duas estratégias de fragmentação, conforme especificado abaixo:

Distribuição igualitária

A estratégia `equal-distribution` de fragmentação divide os testes em compilações paralelas com base na ordem alfabética dos nomes dos arquivos de teste. Essa abordagem primeiro classifica os arquivos de teste e, em seguida, emprega um método baseado em fragmentos para distribuí-los, garantindo que arquivos semelhantes sejam agrupados para teste. É recomendado ao lidar com um conjunto relativamente pequeno de arquivos de teste. Embora esse método tenha como objetivo alocar um número aproximadamente igual de arquivos para cada fragmento, com uma diferença máxima de um, ele não garante estabilidade. Quando arquivos de teste são adicionados ou removidos em compilações subsequentes, a distribuição dos arquivos existentes pode mudar, causando potencialmente a reatribuição entre fragmentos.

Estabilidade

A estratégia `stability` de fragmentação emprega um algoritmo de hash consistente para dividir os testes entre fragmentos, garantindo que a distribuição de arquivos permaneça estável. Quando novos arquivos são adicionados ou removidos, essa abordagem garante que as `file-to-shard` atribuições existentes permaneçam praticamente inalteradas. Para suítes de teste grandes, é recomendável usar a opção de estabilidade para distribuir uniformemente os testes entre os fragmentos. Esse mecanismo visa fornecer uma distribuição quase igual, garantindo que cada fragmento receba um número similar de arquivos, com apenas uma variação mínima. Embora a estratégia de estabilidade não garanta uma distribuição igualitária ideal, ela oferece uma distribuição quase igual que mantém a consistência nas atribuições de arquivos entre compilações, mesmo quando os arquivos são adicionados ou removidos.

Para habilitar a divisão de testes, você precisa configurar a seção de lotes nas configurações do seu CodeBuild projeto, especificando o `parallelism` nível desejado e outros parâmetros relevantes. Além disso, você precisará incluir o `codebuild-tests-run` utilitário em seu arquivo `buildspec`, junto com os comandos de teste e o método de divisão apropriados.

Mescle automaticamente relatórios de criação individuais

Nas compilações de lote do Fanout, AWS CodeBuild suporta a mesclagem automática de relatórios de compilação individuais em um relatório consolidado em nível de lote. Esse recurso fornece uma visão abrangente dos resultados dos testes e da cobertura do código em todas as compilações em um lote.

Como funciona

Ao executar compilações fanout em lote, cada compilação individual gera relatórios de [teste](#). CodeBuild em seguida, consolida automaticamente relatórios idênticos de diferentes compilações em um relatório unificado, que é anexado à criação em lote. Esses relatórios consolidados são facilmente acessíveis por meio do `reportArns` campo da [BatchGetBuildBatches](#) API e também podem ser visualizados na guia Relatórios do console. Esse recurso de mesclagem também se estende aos relatórios descobertos automaticamente.

Os relatórios consolidados são criados em [grupos de relatórios](#) que são especificados no `buildspec` ou descobertos automaticamente por CodeBuild. Você pode analisar as tendências dos relatórios mesclados diretamente nesses grupos de relatórios, fornecendo informações valiosas sobre o desempenho geral da compilação e as métricas de qualidade em compilações históricas do mesmo projeto de compilação em lote.

Para cada compilação individual dentro do lote, o CodeBuild automaticamente cria grupos de relatórios separados. Eles seguem uma convenção de nomenclatura específica, combinando o nome do grupo de relatórios de criação em lote com um sufixo `deBuildFanoutShard<shard_number>`, onde o `shard_number` representa o número do fragmento no qual o grupo de relatórios é criado. Essa organização permite que você acompanhe e analise tendências nos níveis de construção consolidado e individual, oferecendo flexibilidade na forma como você monitora e avalia seus processos de construção.

O relatório de criação em lote segue a mesma estrutura dos relatórios de [criação individuais](#). Os seguintes campos-chave na guia Relatório são específicos para relatórios de criação em lote:

Status do relatório de criação em lote

O status dos relatórios de criação em lote segue regras específicas, dependendo do tipo de relatório:

- Relatórios de teste:
 - Bem-sucedido: o status é definido como bem-sucedido quando todos os relatórios de criação individuais tiverem sido bem-sucedidos.
 - Falha: o status é definido como Falha se algum relatório de compilação individual falhar.
 - Incompleto: o status é marcado como incompleto se algum relatório de compilação individual estiver ausente ou tiver um status incompleto.
- Relatórios de cobertura de código:
 - Concluído: o status é definido como concluído quando todos os relatórios de compilação individuais estiverem concluídos.
 - Falha: o status é definido como Falha se algum relatório de compilação individual falhar.
 - Incompleto: o status é marcado como incompleto se algum relatório de compilação individual estiver ausente ou tiver um status incompleto.

Resumo do teste

O relatório de teste mesclado consolida os seguintes campos de todos os relatórios de criação individuais:

- `duration-in-nano-seconds`: Tempo máximo de duração do teste em nanossegundos entre todos os relatórios de criação individuais.
- `total`: a contagem combinada de todos os casos de teste, somando o número total de testes de cada compilação.
- `contagens de status`: fornece uma visão consolidada dos status do teste, como aprovado, reprovado ou ignorado, calculado agregando a contagem de cada tipo de status em todas as compilações individuais.

Resumo da cobertura do código

O relatório de cobertura de código mesclado combina campos de todas as compilações individuais usando os seguintes cálculos:

- `filiais cobertas`: soma de todas as filiais cobertas a partir de relatórios individuais.
- `ramificações perdidas`: soma de todas as filiais perdidas de relatórios individuais.
- `branch-coverage-percentage`: $(\text{Total covered branches} / \text{Total branches}) * 100$

- linhas cobertas: soma de todas as linhas cobertas de relatórios individuais.
- linhas perdidas: soma de todas as linhas perdidas de relatórios individuais.
- lines-coverage-percentage: $(\text{Total covered lines} / \text{Total lines}) * 100$

ID de execução

O ARN de compilação em lote.

Casos de teste

O relatório mesclado contém uma lista consolidada de todos os casos de teste de compilações individuais, acessível por meio da [DescribeTestCases](#) API e do relatório de criação em lote no console.

Coberturas de código

O relatório de cobertura de código mesclado fornece informações consolidadas de cobertura de linha e ramificação para cada arquivo em todas as compilações individuais, acessíveis por meio da [DescribeCodeCoverages](#) API e do relatório de criação em lote no console. Nota: Para arquivos cobertos por vários arquivos de teste distribuídos em diferentes fragmentos, o relatório mesclado usa os seguintes critérios de seleção:

1. A seleção primária é baseada na maior cobertura de linha entre os fragmentos.
2. Se a cobertura da linha for igual em vários fragmentos, o fragmento com a maior cobertura de ramificação será selecionado.

Exemplo de execução paralela de testes para várias estruturas de teste

Você pode usar o comando `codebuild-tests-run` CLI para dividir e executar seus testes em ambientes de execução paralela. A seção a seguir fornece `buildspec.yml` exemplos de várias estruturas, ilustrando o uso do `codebuild-tests-run` comando.

- Cada exemplo abaixo inclui um `parallelism` nível de cinco, o que significa que cinco ambientes de execução idênticos serão criados para dividir seus testes. Você pode escolher um `parallelism` nível adequado ao seu projeto modificando o `parallelism` valor na `build-fanout` seção.
- Cada exemplo abaixo mostra como configurar seus testes para serem divididos pelo nome do arquivo de teste, que é por padrão. Isso distribui os testes uniformemente nos ambientes de execução paralela.

Antes de começar, consulte [Execute testes paralelos em compilações em lote](#) para obter mais informações.

Para obter uma lista completa das opções ao usar o comando `codebuild-tests-run CLI`, consulte. [Use o comando codebuild-tests-run CLI](#)

Tópicos

- [Configure testes paralelos com o Django](#)
- [Configure testes paralelos com o Elixir](#)
- [Configure testes paralelos com Go](#)
- [Configure testes paralelos com Java \(Maven\)](#)
- [Configure testes paralelos com Javascript \(Jest\)](#)
- [Configure testes paralelos com o Kotlin](#)
- [Configure testes paralelos com PHPUnit](#)
- [Configure testes paralelos com o Pytest](#)
- [Configure testes paralelos com Ruby \(Cucumber\)](#)
- [Configure testes paralelos com Ruby \(\) RSpec](#)

Configure testes paralelos com o Django

A seguir está uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com o Django em uma plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - sudo yum install -y python3 python3-pip
      - python3 -m ensurepip --upgrade
      - python3 -m pip install django
```

```

pre_build:
  commands:
    - echo 'Prebuild'
build:
  commands:
    - echo 'Running Django Tests'
    - |
      codebuild-tests-run \
        --test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES"
| sed -E "s/\//__/g; s/\.py$/; s/__/./g")' \
        --files-search "codebuild-glob-search '**/tests/*test_*.py'" \
        --sharding-strategy 'equal-distribution'
post_build:
  commands:
    - echo 'Test execution completed'

```

O exemplo acima mostra o uso da variável de ambiente `CODEBUILD_CURRENT_SHARD_FILES`. Aqui `CODEBUILD_CURRENT_SHARD_FILES` é usado para buscar caminhos de arquivo de notação de pontos suportados pelo Django. Use `CODEBUILD_CURRENT_SHARD_FILES` entre aspas duplas, conforme mostrado acima.

Configure testes paralelos com o Elixir

A seguir está uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com o Elixir em uma plataforma Ubuntu:

```

version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Elixir dependencies'
      - sudo apt update
      - sudo DEBIAN_FRONTEND=noninteractive apt install -y elixir
      - elixir --version
      - mix --version
  pre_build:
    commands:

```

```
- echo 'Prebuild'
build:
  commands:
    - echo 'Running Elixir Tests'
    - |
      codebuild-tests-run \
        --test-command 'mix test' \
        --files-search "codebuild-glob-search '**/test/**/*_test.exs'" \
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Configure testes paralelos com Go

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com Go em uma plataforma Linux:

```
version: 0.2

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

phases:
  install:
    commands:
      - echo 'Fetching Go version'
      - go version
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running go Tests'
      - go mod init calculator
      - cd calc
      - |
        codebuild-tests-run \
          --test-command "go test -v calculator.go" \
          --files-search "codebuild-glob-search '**/*test.go'"

```

```
post_build:
  commands:
    - echo "Test execution completed"
```

No exemplo acima, a `calculator.go` função contém funções matemáticas simples para testar e todos os arquivos e `calculator.go` arquivos de teste estão dentro da `calc` pasta.

Configure testes paralelos com Java (Maven)

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com Java em uma plataforma Linux:

```
version: 0.2

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

phases:
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo "Running mvn test"
      - |
        codebuild-tests-run \
          --test-command 'mvn test -Dtest=$(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
"s|src/test/java/||g; s|\..java//g; s|/|.|g; s/ /,/g" | tr "\n" "," | sed "s/,,$//")' \
          --files-search "codebuild-glob-search '**/test/**/*.java'"
  post_build:
    commands:
      - echo "Running post-build steps..."
      - echo "Test execution completed"
```

No exemplo fornecido, a variável de ambiente `CODEBUILD_CURRENT_SHARD_FILES` contém arquivos de teste no fragmento atual, separados por novas linhas. Esses arquivos são convertidos em uma lista separada por vírgulas de nomes de classes no formato aceito pelo `-Dtest` parâmetro para Maven.

Configure testes paralelos com Javascript (Jest)

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com Javascript em uma plataforma Ubuntu:

```
version: 0.2

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

phases:
  install:
    commands:
      - echo 'Installing Node.js dependencies'
      - apt-get update
      - apt-get install -y nodejs
      - npm install
      - npm install --save-dev jest-junit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running JavaScript Tests'
      - |
        codebuild-tests-run \
          --test-command "npm run test" \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'stability'
  post_build:
    commands:
      - echo 'Test execution completed'
```

Configure testes paralelos com o Kotlin

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com Kotlin em uma plataforma Linux:

```
version: 0.2
```

```

batch:
  fast-fail: false
  build-fanout:
    parallelism: 2
    ignore-failure: false

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - echo 'Installing dependencies'
      - KOTLIN_VERSION="1.8.20" # Replace with your desired version
      - curl -o kotlin-compiler.zip -L "https://github.com/JetBrains/kotlin/releases/
download/v${KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip"
      - unzip kotlin-compiler.zip -d /usr/local
      - export PATH=$PATH:/usr/local/kotlinc/bin
      - kotlin -version
      - curl -O https://repo1.maven.org/maven2/org/junit/platform/junit-platform-
console-standalone/1.8.2/junit-platform-console-standalone-1.8.2.jar
    pre_build:
      commands:
        - echo 'prebuild'
    build:
      commands:
        - echo 'Running Kotlin Tests'
        - |
          codebuild-tests-run \
            --test-command 'kotlinc src/main/kotlin/*.kt $(echo
"$CODEBUILD_CURRENT_SHARD_FILES" | tr "\n" " ") -d classes -cp junit-platform-console-
standalone-1.8.2.jar' \
            --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
        - |
          codebuild-tests-run \
            --test-command '
          java -jar junit-platform-console-standalone-1.8.2.jar --class-path classes
\
          $(for file in $CODEBUILD_CURRENT_SHARD_FILES; do
            class_name=$(basename "$file" .kt)
            echo "--select-class $class_name"
          done)
          ' \
            --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
    post_build:

```

```
commands:
  - echo "Test execution completed"
```

No exemplo acima, a `codebuild-tests-run` CLI é usada duas vezes. Durante a primeira execução, o `kotlinc` compila os arquivos. A `CODEBUILD_CURRENT_SHARD_FILES` variável recupera os arquivos de teste atribuídos ao fragmento atual, que são então convertidos em uma lista separada por espaços. Na segunda execução, JUnit executa os testes. Novamente, `CODEBUILD_CURRENT_SHARD_FILES` busca os arquivos de teste atribuídos ao fragmento atual, mas dessa vez eles são convertidos em nomes de classe.

Configure testes paralelos com PHPUnit

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes PHPUnit em uma plataforma Linux:

```
version: 0.2

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

phases:
  install:
    commands:
      - echo 'Install dependencies'
      - composer require --dev phpunit/phpunit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running phpunit Tests'
      - composer dump-autoload
      - |
        codebuild-tests-run \
          --test-command "./vendor/bin/phpunit --debug" \
          --files-search "codebuild-glob-search '**/tests/*Test.php'"
  post_build:
    commands:
      - echo 'Test execution completed'
```

Configure testes paralelos com o Pytest

A seguir está uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com o Pytest em uma plataforma Ubuntu:

```
version: 0.2

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

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - apt-get update
      - apt-get install -y python3 python3-pip
      - pip3 install --upgrade pip
      - pip3 install pytest
  build:
    commands:
      - echo 'Running Python Tests'
      - |
        codebuild-tests-run \
          --test-command 'python -m pytest' \
          --files-search "codebuild-glob-search 'tests/test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

A seguir está um exemplo de um `buildspec.yml` que mostra a execução paralela do teste com o Pytest em uma plataforma Windows:

```
version: 0.2

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



```

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - pip install pytest
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running pytest'
      - |
        & codebuild-tests-run `
        --test-command 'pytest @("$env:CODEBUILD_CURRENT_SHARD_FILES" -split \"`r?`n
        \")' `
        --files-search "codebuild-glob-search '**/test_*.py' '**/*_test.py'" `
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"

```

No exemplo acima, a variável de `CODEBUILD_CURRENT_SHARD_FILES` ambiente é usada para buscar arquivos de teste atribuídos ao fragmento atual e passados como matriz para o comando `pytest`.

Configure testes paralelos com Ruby (Cucumber)

Veja a seguir uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes com o Cucumber em uma plataforma Linux:

```

version: 0.2

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

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'

```

```
- gem install bundler
- bundle install
pre_build:
  commands:
    - echo 'prebuild'
build:
  commands:
    - echo 'Running Cucumber Tests'
    - cucumber --init
    - |
      codebuild-tests-run \
        --test-command "cucumber" \
        --files-search "codebuild-glob-search '**/*.feature'"
post_build:
  commands:
    - echo "Test execution completed"
```

Configure testes paralelos com Ruby () RSpec

A seguir está uma amostra de uma `buildspec.yml` que mostra a execução paralela de testes RSpec em uma plataforma Ubuntu:

```
version: 0.2

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

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - apt-get update
      - apt-get install -y ruby ruby-dev build-essential
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo 'Running Ruby Tests'
      - |
        codebuild-tests-run \
```

```
--test-command 'bundle exec rspec' \  
--files-search "codebuild-glob-search 'spec/**/*.rb'" \  
--sharding-strategy 'equal-distribution'  
post_build:  
  commands:  
    - echo "Test execution completed"
```

Compilações em cache para melhorar o desempenho

Você pode economizar tempo quando seu projeto é compilado usando um cache. Um cache pode armazenar partes reutilizáveis do seu ambiente de build e usá-las em vários builds. O projeto de compilação pode usar um dos dois tipos de armazenamento em cache: Amazon S3 ou local. Se usar um cache local, você deverá escolher um ou mais dos três modos de cache: cache de origem, cache de camada do Docker e cache personalizado.

Note

O modo de cache de camada do Docker está disponível somente para o ambiente Linux. Se você escolher esse modo, deverá executar sua compilação no modo privilegiado. CodeBuild projetos com modo privilegiado concedem ao contêiner acesso a todos os dispositivos. Para obter mais informações, consulte [Runtime privilege and Linux capabilities](#) no site de documentação do Docker.

Tópicos

- [Armazenamento em cache do Amazon S3](#)
- [Armazenamento em cache local](#)
- [Especificar um cache local](#)

Armazenamento em cache do Amazon S3

O armazenamento em cache do Amazon S3 armazena o cache em um bucket do Amazon S3 que está disponível em vários hosts de compilação. Esta é uma boa opção para artefatos de compilação pequenos a intermediários que são mais caras para criar do que para baixar.

Para usar o Amazon S3 em uma compilação, você pode especificar os caminhos para os arquivos nos quais deseja armazenar em cache. `buildspec.yml` CodeBuild armazenará e atualizará

automaticamente o cache na localização do Amazon S3 configurada no projeto. Se você não especificar os caminhos do arquivo, CodeBuild fará o melhor possível para armazenar em cache as dependências de linguagem comum para ajudá-lo a acelerar as compilações. Você pode ver os detalhes do cache nos registros de compilação.

Além disso, se quiser ter várias versões de cache, você pode definir uma chave de cache no `buildspec.yml`. CodeBuild armazena o cache no contexto dessa chave de cache e cria uma cópia de cache exclusiva que não será atualizada depois de criada. As chaves de cache também podem ser compartilhadas entre projetos. Recursos como chaves dinâmicas, controle de versão de cache e compartilhamento de cache entre compilações só estão disponíveis quando uma chave é especificada.

Para saber mais sobre a sintaxe do cache no arquivo `buildspec`, consulte [cache](#) a referência `buildspec`.

Tópicos

- [Gere chaves dinâmicas](#)
- [codebuild-hash-files](#)
- [Versão de cache](#)
- [Compartilhamento de cache entre projetos](#)
- [Exemplos de Buildspec](#)

Gere chaves dinâmicas

Uma chave de cache pode incluir comandos de shell e variáveis de ambiente para torná-la única, permitindo atualizações automáticas de cache quando a chave é alterada. Por exemplo, você pode definir uma chave usando o hash do `package-lock.json` arquivo. Quando as dependências nesse arquivo mudam, o hash — e, portanto, a chave do cache — muda, acionando a criação automática de um novo cache.

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

CodeBuild avaliará a expressão `$(codebuild-hash-files package-lock.json)` para obter a chave final:

```
npm-key-abc123
```

Você também pode definir uma chave de cache usando variáveis de ambiente, como `CODEBUILD_RESOLVED_SOURCE_VERSION`. Isso garante que sempre que sua fonte mudar, uma nova chave seja gerada, resultando em um novo cache sendo salvo automaticamente:

```
cache:
  key: npm-key-${CODEBUILD_RESOLVED_SOURCE_VERSION}
```

CodeBuild avaliará a expressão e obterá a chave final:

```
npm-key-046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

codebuild-hash-files

`codebuild-hash-files` é uma ferramenta de CLI que calcula um hash SHA-256 para um conjunto de arquivos no diretório de origem usando padrões globais: CodeBuild

```
codebuild-hash-files <glob-pattern-1> <glob-pattern-2> ...
```

Aqui estão alguns exemplos de uso de `codebuild-hash-files`:

```
codebuild-hash-files package-lock.json
codebuild-hash-files '**/*.md'
```

Versão de cache

A versão em cache é um hash gerado a partir dos caminhos dos diretórios que estão sendo armazenados em cache. Se dois caches tiverem versões diferentes, eles serão tratados como caches distintos durante o processo de correspondência. Por exemplo, os dois caches a seguir são considerados diferentes porque fazem referência a caminhos diferentes:

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target pip-dependencies
cache:
  key: pip-dependencies
```

```
paths:
  - "pip-dependencies/**/*"
```

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target tmp/pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "tmp/pip-dependencies/**/*"
```

Compartilhamento de cache entre projetos

Você pode usar o campo `cacheNamespace` API na cache seção para compartilhar um cache em vários projetos. Esse campo define o escopo do cache. Para compartilhar um cache, faça o seguinte:

- Use o mesmo `cacheNamespace`.
- Especifique o mesmo `cachekey`.
- Defina caminhos de cache idênticos.
- Use os mesmos buckets do Amazon S3 `basePathPrefix`, se configurados.

Isso garante a consistência e permite o compartilhamento de cache entre projetos.

Especifique um namespace de cache (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Escolha Criar projeto. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
3. Em Artefatos, escolha Configuração adicional.
4. Para Tipo de cache, escolha Amazon S3.
5. Em Cache namespace - opcional, insira um valor de namespace.

▼ Additional configuration

Cache, encryption key

Encryption key - optional

Provide the AWS KMS customer master key used to encrypt this build's output artifacts. The default is your AWS-managed customer master key for S3.

arn:aws:kms:<region-ID>:<account-ID>:key/<key-ID>

Cache type

Amazon S3 ▼

Cache bucket**Cache path prefix - optional****Cache lifecycle (days) - optional**

You can apply a lifecycle expiration action to all or a subset of objects in the cache bucket based on the path prefix.

[+ Add expiration](#)**Cache namespace - optional**

test-cache-namespace

Provide a cache namespace if you want to share caches across projects.

6. Continue com os valores padrão e escolha Criar projeto de compilação.

Especifique um namespace de cache (AWS CLI)

Você pode usar o `--cache` parâmetro no AWS CLI para especificar um namespace de cache.

```
--cache '{"type": "S3", "location": "your-s3-bucket", "cacheNamespace": "test-cache-namespace"}'
```

Exemplos de Buildspec

Aqui estão vários exemplos de buildspec para linguagens comuns:

Tópicos

- [Armazene em cache as dependências do Node.js](#)

- [Dependências de cache do Python](#)
- [Dependências do Cache Ruby](#)
- [Dependências do Cache Go](#)

Armazene em cache as dependências do Node.js

Se seu projeto inclui um `package-lock.json` arquivo e usa npm para gerenciar dependências do Node.js, o exemplo a seguir mostra como configurar o armazenamento em cache. Por padrão, npm instala dependências no `node_modules` diretório.

```
version: 0.2

phases:
  build:
    commands:
      - npm install
cache:
  key: npm-${codebuild-hash-files package-lock.json}
  paths:
    - "node_modules/**/*"
```

Dependências de cache do Python

Se seu projeto inclui um `requirements.txt` arquivo e usa pip para gerenciar dependências do Python, o exemplo a seguir demonstra como configurar o armazenamento em cache. Por padrão, o pip instala pacotes no diretório do `site-packages` sistema.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt
cache:
  key: python-${codebuild-hash-files requirements.txt}
  paths:
    - "/root/.pyenv/versions/${python_version}/lib/python${python_major_version}/site-packages/**/*"
```


Além disso, você pode instalar dependências em um diretório específico e configurar o armazenamento em cache para esse diretório.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt --target python-dependencies
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "python-dependencies/**/*"
```

Dependências do Cache Ruby

Se seu projeto inclui um `Gemfile.lock` arquivo e o usa Bundler para gerenciar dependências de gem, o exemplo a seguir demonstra como configurar o armazenamento em cache de forma eficaz.

```
version: 0.2

phases:
  build:
    commands:
      - bundle install --path vendor/bundle
cache:
  key: ruby-$(codebuild-hash-files Gemfile.lock)
  paths:
    - "vendor/bundle/**/*"
```

Dependências do Cache Go

Se seu projeto inclui um `go.sum` arquivo e usa módulos Go para gerenciar dependências, o exemplo a seguir demonstra como configurar o armazenamento em cache. Por padrão, os módulos Go são baixados e armazenados no `${GOPATH}/pkg/mod` diretório.

```
version: 0.2

phases:
  build:
    commands:
```

```
- go mod download
cache:
  key: go-$(codebuild-hash-files go.sum)
  paths:
    - "/go/pkg/mod/**/*"
```

Armazenamento em cache local

O armazenamento em cache local armazena um cache localmente em um host de compilação que está disponível somente para esse host de compilação. Esta é uma boa opção para artefatos de compilação grandes e intermediários porque o cache fica imediatamente disponível no host de compilação. Essa não é a melhor opção se suas compilações são pouco frequentes. Isso significa que o desempenho da compilação não é afetado pelo tempo de transferência na rede.

Se optar por armazenamento em cache local, você deverá escolher um ou mais dos seguintes modos de cache:

- O modo de cache de origem armazena em cache os metadados do Git para origens principal e secundária. Depois que o cache é criado, as compilações subsequentes extraem somente a alteração entre as confirmações. Esse modo é uma boa opção para projetos com um diretório de trabalho limpo e uma origem que seja um grande repositório Git. Se você escolher essa opção e seu projeto não usar um repositório Git (AWS CodeCommit, GitHub, GitHub Enterprise Server ou Bitbucket), a opção será ignorada.
- O modo de cache da camada do Docker armazena as camadas do Docker existentes. Esse modo é uma boa opção para projetos que compilam ou extraem grandes imagens do Docker. Ele pode impedir os problemas de desempenho causados pela extração de grandes imagens do Docker da rede.

Note

- Só é possível usar um cache de camada do Docker no ambiente Linux.
- O sinalizador `privileged` deve ser definido para que o projeto tenha as permissões necessárias do Docker.

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

- Você deve considerar as implicações de segurança antes de usar um cache de camada do Docker.
-
- O modo de cache personalizado armazena em cache os diretórios que você especificar no arquivo buildspec. Esse modo é uma boa opção quando o cenário de compilação não é adequado para um dos dois outros modos de cache local. Se você usar um cache personalizado:
 - Somente diretórios pode ser especificado para o armazenamento em cache. Não é possível especificar arquivos individuais.
 - Symlinks são usados para fazer referência a diretórios armazenados em cache.
 - Os diretórios em cache são vinculados à sua compilação antes que ela faça download de suas origens do projeto. Itens armazenados em cache substituem itens de origem se tiverem o mesmo nome. Os diretórios são especificados usando caminhos de cache no arquivo buildspec. Para obter mais informações, consulte [Sintaxe de buildspec](#).
 - Evite nomes de diretório iguais na origem e no cache. Os diretórios armazenados em cache local podem substituir ou excluir o conteúdo de diretórios no repositório de origem que tenham o mesmo nome.

Note

O armazenamento em cache local não é compatível com o tipo de ambiente LINUX_GPU_CONTAINER e o tipo de computação BUILD_GENERAL1_2XLARGE. Para obter mais informações, consulte [Modos e tipos de computação do ambiente de compilação](#).

Note

O armazenamento em cache local não é suportado quando você configura CodeBuild para trabalhar com uma VPC. Para obter mais informações sobre como usar VPCs com CodeBuild, consulte [Use AWS CodeBuild com a Amazon Virtual Private Cloud](#).

Especificar um cache local

Você pode usar o console AWS CLI, o SDK ou AWS CloudFormation especificar um cache local. Para obter mais informações sobre armazenamento em cache local, consulte [Armazenamento em cache local](#).

Tópicos

- [Especificar o armazenamento em cache local \(CLI\)](#)
- [Especificar o armazenamento em cache local \(console\)](#)
- [Especificar o armazenamento em cache local \(AWS CloudFormation\)](#)

Especificar o armazenamento em cache local (CLI)

Você pode usar o `--cache` parâmetro no AWS CLI para especificar cada um dos três tipos de cache local.

- Para especificar um cache de origem:

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Para especificar um cache de camada do Docker:

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

- Para especificar um cache personalizado:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).

Especificar o armazenamento em cache local (console)

Você especifica um cache na seção Artifacts (Artefatos) do console. Em Tipo de cache, selecione Amazon S3 ou Local. Se você optar por Local, selecione uma ou mais das três opções de cache local.

Cache type

Local ▼

Select one or more local cache options.

- ☐ **Docker layer cache**
Caches existing Docker layers so they can be reused. Requires privileged mode.
- ☐ **Source cache**
Caches .git metadata so subsequent builds only pull the change in commits.
- ☐ **Custom cache**
Caches directories specified in the buildspec file.

Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#).

Especificar o armazenamento em cache local (AWS CloudFormation)

Se você usa AWS CloudFormation para especificar um cache local, na Cache propriedade, forType, especifiqueLOCAL. O exemplo de AWS CloudFormation código formatado em YAML a seguir especifica todos os três tipos de cache local. Você pode especificar qualquer combinação de tipos. Se você usar um cache de camada do Docker, em Environment, defina PrivilegedMode como true e Type como LINUX_CONTAINER.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: <service-role>
    Artifacts:
      Type: S3
      Location: <bucket-name>
      Name: myArtifact
      EncryptionDisabled: true
      OverrideArtifactName: true
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
      Certificate: <bucket/cert.zip>
      # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
      PrivilegedMode: true
    Source:
```

```
Type: GITHUB
Location: <github-location>
InsecureSsl: true
GitCloneDepth: 1
ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
  Type: LOCAL
  Modes: # You can specify one or more cache mode,
    - LOCAL_CUSTOM_CACHE
    - LOCAL_DOCKER_LAYER_CACHE
    - LOCAL_SOURCE_CACHE
```

Note

Por padrão, o daemon do Docker está habilitado para compilações não VPC. Se você quiser usar contêineres do Docker para compilações da VPC, consulte [Privilegio de tempo de execução e funcionalidades do Linux](#) no site do Docker Docs e ative o modo privilegiado. Além disso, o Windows não é compatível com o modo privilegiado.

Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CloudFormation\)](#).

Compilações de depuração AWS CodeBuild

AWS CodeBuild fornece dois métodos para depurar compilações durante o desenvolvimento e a solução de problemas. Você pode usar o ambiente CodeBuild Sandbox para investigar problemas e validar correções em tempo real, ou você pode usar o AWS Systems Manager Session Manager para se conectar ao contêiner de compilação e visualizar o estado do contêiner.

Depure compilações com sandbox CodeBuild

O ambiente CodeBuild sandbox fornece uma sessão de depuração interativa em um ambiente seguro e isolado. Você pode interagir com o ambiente diretamente por meio do AWS Management Console or AWS CLI, executar comandos e validar seu processo de criação passo a passo. Ele usa um modelo econômico de cobrança por segundo e oferece suporte à mesma integração nativa com provedores e AWS serviços de origem que seu ambiente de construção. Você também pode se conectar a um ambiente sandbox usando clientes SSH ou de seus ambientes de desenvolvimento integrados (IDEs).

Para saber mais sobre os preços do CodeBuild sandbox, acesse a [documentação CodeBuild de preços](#). Para obter instruções detalhadas, acesse a [Depure compilações com sandbox CodeBuild documentação](#).

Depurar compilações com o Session Manager

AWS O Systems Manager Session Manager permite acesso direto às compilações em execução em seu ambiente de execução real. Essa abordagem permite que você se conecte a contêineres de compilação ativos e inspecione o processo de criação em tempo real. Você pode examinar o sistema de arquivos, monitorar os processos em execução e solucionar problemas à medida que eles ocorrem.

Para obter instruções detalhadas, acesse a [Depurar compilações com o Gerenciador de Sessões documentação](#).

Depure compilações com sandbox CodeBuild

Em AWS CodeBuild, você pode depurar uma compilação usando o CodeBuild sandbox para executar comandos personalizados e solucionar problemas de sua compilação.

Tópicos

- [Pré-requisitos](#)
- [Depurar compilações com CodeBuild sandbox \(console\)](#)
- [Depure compilações com CodeBuild sandbox \(\)AWS CLI](#)
- [Tutorial: conectando-se a uma sandbox usando SSH](#)
- [Solução de problemas de conexão SSH do AWS CodeBuild sandbox](#)

Pré-requisitos

Antes de usar um CodeBuild sandbox, certifique-se de que sua função CodeBuild de serviço tenha a seguinte política de SSM:

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

```
"Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
    "ssmmessages:OpenDataChannel"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:StartSession"
    ],
    "Resource": [
        "arn:aws:codebuild:<region>:<account-id>:build/*",
        "arn:aws:ssm:<region>::document/AWS-StartSSHSession"
    ]
}
]
```

Depurar compilações com CodeBuild sandbox (console)

Use as instruções a seguir para executar comandos e conectar seu cliente SSH ao CodeBuild sandbox no console.

Execute comandos com o CodeBuild sandbox (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação. Escolha o projeto de compilação e, em seguida, escolha Debug build.

sandbox-project Actions ▼ Create trigger Edit Clone Clear cache Debug build Start build with overrides **Start build**

Configuration

Source provider No source	Primary repository -	Artifacts upload location -	Service role arn:aws:iam::[redacted]:role/service-role/codebuild-sandbox-project-service-role
Public builds Disabled			

Build history Batch history **Project details** Build triggers Metrics Debug sessions

Project configuration Edit

Name sandbox-project	Description -
Project ARN ☐ arn:aws:codebuild:us-east-1:[redacted]:project/sandbox-project	Build badge Disabled

3. Na guia Executar comando, insira seus comandos personalizados e escolha Executar comando.

Debug build

Run Command SSH Client Session Manager

Run custom commands with sandbox Learn more

- Launches a sandbox environment mirroring your project configuration.
- Automatically downloads source code, while skipping project buildspec execution.
- Ideal for reproducing failure, experimenting fixes and investigation.

Command

```
1 pwd
```

⊗ ⊕ 0 1:4 SH

Run command

4. Sua CodeBuild sandbox será então inicializada e começará a executar seus comandos personalizados. A saída será mostrada na guia Saída quando for concluída.

Debug build

[Run Command](#)[SSH Client](#)[Session Manager](#)**Sandbox is running**Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is ready and available for use.[Stop sandbox](#)

Command

1 `pwd`

⊞ ⊞

1:1 SH

[Run command](#)[Command output](#)[Sandbox phases](#)[Sandbox logs](#)[Sandbox configurations](#)[Command history](#)[View entire log in CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

5. Quando a solução de problemas estiver concluída, você poderá interromper a sandbox escolhendo Parar sandbox. Em seguida, escolha Parar para confirmar que sua sandbox será interrompida.

Stop sandbox



Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

- `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18`

[Cancel](#)[Stop](#)

Debug build

[Run Command](#)[SSH Client](#)[Session Manager](#)**Sandbox is stopped**Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is currently inactive.[Command output](#)[Sandbox phases](#)[Sandbox logs](#)[Sandbox configurations](#)[Command history](#)[View entire log in CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

Conecte-se ao seu cliente SSH com CodeBuild sandbox (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação. Escolha o projeto de compilação e, em seguida, escolha Debug build.

sandbox-project [Actions ▼](#) [Create trigger](#) [Edit](#) [Clone](#) [Clear cache](#) [Debug build](#) [Start build with overrides](#) [Start build](#)

Configuration

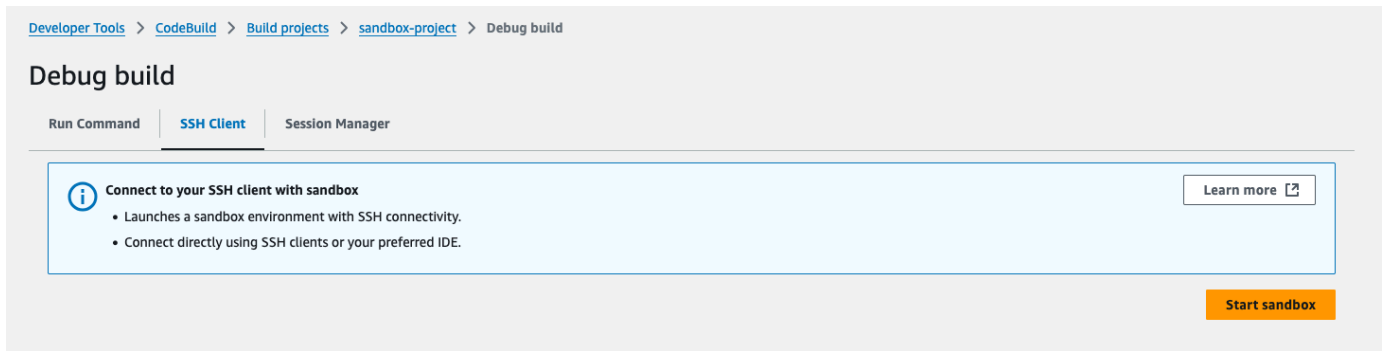
Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	<code>arn:aws:iam:::role/service-role/codebuild-sandbox-project-service-role</code>
Public builds	Disabled		

[Build history](#) [Batch history](#) [Project details](#) [Build triggers](#) [Metrics](#) [Debug sessions](#)

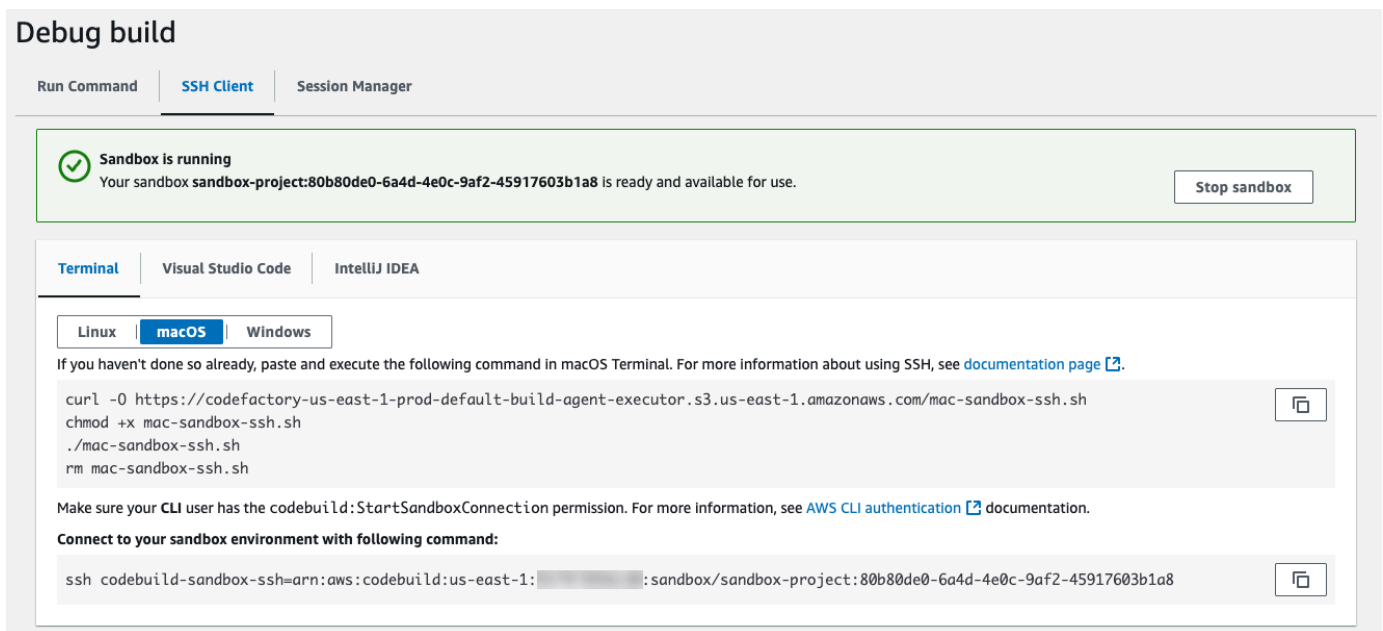
Project configuration [Edit](#)

Name	Description
sandbox-project	-
Project ARN	Build badge
☐ <code>arn:aws:codebuild:us-east-1:::project/sandbox-project</code>	Disabled

3. Na guia Cliente SSH, escolha Iniciar sandbox.



4. Depois que a CodeBuild sandbox começar a ser executada, siga as instruções do console para conectar seu cliente SSH à sandbox.



5. Quando a solução de problemas estiver concluída, você poderá interromper a sandbox escolhendo Parar sandbox. Em seguida, escolha Parar para confirmar que sua sandbox será interrompida.

Stop sandbox

×

Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

• sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8

Cancel

Stop

Debug build

Run CommandSSH ClientSession Manager

⚠

Sandbox is stopped

Your sandbox `sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8` is currently inactive.

Sandbox phasesSandbox logsSandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
QUEUED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
PROVISIONING	✔ Succeeded		5 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	✔ Succeeded		8 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
RUN_SANDBOX	✔ Succeeded		213 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
UPLOAD_ARTIFACTS	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
FINALIZING	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
COMPLETED	✔ Succeeded	-	-	Apr 8, 2025 1:36 PM (UTC-7:00)	-

Depure compilações com CodeBuild sandbox ()AWS CLI

Use as instruções a seguir para executar comandos e conectar seu cliente SSH ao CodeBuild sandbox.

CodeBuild Inicie uma sandbox ()AWS CLI

CLI command

```
aws codebuild start-sandbox --project-name $PROJECT_NAME
```

- `--project-name`: nome CodeBuild do projeto

Sample request

```
aws codebuild start-sandbox --project-name "project-name"
```

Sample response

```
{
  "id": "project-name",
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
  "projectName": "project-name",
  "requestTime": "2025-02-06T11:24:15.560000-08:00",
  "status": "QUEUED",
  "source": {
    "type": "S3",
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-sources/eb-sample-jetty-v4.zip",
    "insecureSsl": false
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:6.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [{
      "name": "foo",
      "value": "bar",
      "type": "PLAINTEXT"
    },
    {
      "name": "bar",
      "value": "baz",
      "type": "PLAINTEXT"
    }
  ],
  "privilegedMode": false,
  "imagePullCredentialsType": "CODEBUILD"
},
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
```

```

        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "QUEUED",
    "status": "QUEUED",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "logs": {
        "groupName": "group",
        "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
            "encryptionDisabled": false
        }
    }
}
}

```

```
}
```

Obtenha informações sobre o status do sandbox ()AWS CLI

CLI command

```
aws codebuild batch-get-sandboxes --ids $SANDBOX_IDS
```

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

- `--ids`: lista separada por vírgula de sandboxIds ou sandboxArns.

Você pode fornecer uma ID de sandbox ou um ARN de sandbox:

- ID do sandbox: `<codebuild-project-name>:<UUID>`

Por exemplo, `.project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`

- ARN da área restrita: `arn:aws:codebuild: <region>: <account-id> <codebuild-project-name> <UUID>`

Por exemplo, `.arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`

Sample response

```
{  
  "sandboxes": [{  
    "id": "project-name",  
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
    "projectName": "project-name",  
    "requestTime": "2025-02-06T11:24:15.560000-08:00",  
    "endTime": "2025-02-06T11:39:21.587000-08:00",  
    "status": "STOPPED",  
    "source": {  
      "type": "S3",
```



```

        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
            "name": "foo",
            "value": "bar",
            "type": "PLAINTEXT"
        },
        {
            "name": "bar",
            "value": "baz",
            "type": "PLAINTEXT"
        }
    ],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"timeoutInMinutes": 10,
"queuedTimeoutInMinutes": 480,
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/
SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "COMPLETED",
    "status": "STOPPED",

```

```
"startTime": "2025-02-06T11:24:15.626000-08:00",
"endTime": "2025-02-06T11:39:21.600000-08:00",
"phases": [{
  "phaseType": "SUBMITTED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.577000-08:00",
  "endTime": "2025-02-06T11:24:15.606000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "QUEUED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.606000-08:00",
  "endTime": "2025-02-06T11:24:16.067000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "PROVISIONING",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:16.067000-08:00",
  "endTime": "2025-02-06T11:24:20.519000-08:00",
  "durationInSeconds": 4,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "DOWNLOAD_SOURCE",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:20.519000-08:00",
  "endTime": "2025-02-06T11:24:22.238000-08:00",
  "durationInSeconds": 1,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "RUNNING_SANDBOX",
  "phaseStatus": "TIMED_OUT",
  "startTime": "2025-02-06T11:24:22.238000-08:00",
  "endTime": "2025-02-06T11:39:21.560000-08:00",
  "durationInSeconds": 899,
```

```

        "contexts": [{
            "statusCode": "BUILD_TIMED_OUT",
            "message": "Build has timed out. "
        }]
    },
    {
        "phaseType": "COMPLETED",
        "startTime": "2025-02-06T11:39:21.560000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}
}],
"sandboxesNotFound": []
}

```

Pare uma caixa de areia ()AWS CLI

CLI command

```
aws codebuild stop-sandbox --id $SANDBOX-ID
```

- `--id`: A sandboxId ou sandboxArn.

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

Sample response

```
{  
  "id": "project-name",  
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
  "projectName": "project-name",  
  "requestTime": "2025-02-06T11:24:15.560000-08:00",  
  "status": "STOPPING",  
  "source": {  
    "type": "S3",  
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-  
sources/eb-sample-jetty-v4.zip",  
    "insecureSsl": false  
  },  
  "environment": {  
    "type": "LINUX_CONTAINER",  
    "image": "aws/codebuild/standard:6.0",  
    "computeType": "BUILD_GENERAL1_SMALL",  
    "environmentVariables": [{  
      "name": "foo",  
      "value": "bar",  
      "type": "PLAINTEXT"  
    },  
    {  
      "name": "bar",  
      "value": "baz",  
      "type": "PLAINTEXT"  
    }  
  ],  
}
```

```
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
  },
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
      "status": "ENABLED",
      "groupName": "group",
      "streamName": "stream"
    },
    "s3Logs": {
      "status": "ENABLED",
      "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
      "encryptionDisabled": false
    }
  },
  "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
  "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
  "currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "RUN_SANDBOX",
    "status": "STOPPING",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "phases": [{
      "phaseType": "SUBMITTED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.144000-08:00",
      "endTime": "2025-02-08T14:33:26.173000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "QUEUED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.173000-08:00",
      "endTime": "2025-02-08T14:33:26.702000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "PROVISIONING",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.702000-08:00",
      "endTime": "2025-02-08T14:33:30.530000-08:00",
      "durationInSeconds": 3,
```

```

        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "DOWNLOAD_SOURCE",
        "phaseStatus": "SUCCEEDED",
        "startTime": "2025-02-08T14:33:30.530000-08:00",
        "endTime": "2025-02-08T14:33:33.478000-08:00",
        "durationInSeconds": 2,
        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "RUN_SANDBOX",
        "startTime": "2025-02-08T14:33:33.478000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
}

```

```
    }  
  }  
}
```

Iniciar a execução de um comando (AWS CLI)

CLI command

```
aws codebuild start-command-execution --command $COMMAND --type $TYPE --sandbox-id  
$SANDBOX-ID
```

- `--command`: O comando que precisa ser executado.
- `--sandbox-id`: A `sandboxId` ou `sandboxArn`.
- `--type`: O tipo de comando, `SHELL`.

Sample request

```
aws codebuild start-command-execution --command "echo \"Hello World\"" --type SHELL --  
sandbox-id "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{  
  "id": "e1c658c2-02bb-42a8-9abb-94835241fcd6",  
  "sandboxId": "f7126a4a-b0d5-452f-814c-fea73718f805",  
  "submitTime": "2025-02-06T20:12:02.683000-08:00",  
  "status": "SUBMITTED",  
  "command": "echo \"Hello World\"",  
  "type": "SHELL",  
  "logs": {  
    "groupName": "group",  
    "streamName": "stream",  
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-  
west-2#logsV2:log-groups/log-group/group/log-events/stream",  
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-  
pool-1-us-west-2-beta-default-build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz?  
region=us-west-2",  
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-  
group:group:log-stream:stream",  
  }  
}
```

```

    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-
build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}

```

Obtenha informações sobre as execuções do comando ()AWS CLI

CLI command

```
aws codebuild batch-get-command-executions --command-execution-ids $COMMAND-IDS --
sandbox-id $SANDBOX-IDS
```

- `--command-execution-ids`: Lista separada por vírgula de `commandExecutionIds`
- `--sandbox-id`: A `sandboxId` ou `sandboxArn`.

Sample request

```
aws codebuild batch-get-command-executions --command-execution-
ids "c3c085ed-5a8f-4531-8e95-87d547f27ffd" --sandbox-id "arn:aws:codebuild:us-
west-2:962803963624:sandbox/project-name"
```

Sample response

```

{
  "commandExecutions": [{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",

```



```

    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
      }
    },
    "commandExecutionsNotFound": []
  }
}

```

Listar as execuções de comandos para uma sandbox (AWS CLI)

CLI command

```
aws codebuild list-command-executions-for-sandbox --sandbox-id $SANDBOX-ID --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --sort-order $SORT_ORDER
```

- `--next-token`: o próximo token, se houver, para obter resultados paginados. Você obterá esse valor da execução anterior das caixas de proteção da lista.

- `--max-results`: (Opcional) O número máximo de registros de sandbox a serem recuperados.
- `--sort-order`: a ordem na qual os registros do sandbox devem ser recuperados.

Sample request

```
aws codebuild list-command-executions-for-sandbox --sandbox-id
"arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
  "commandExecutions": [{
    "id": "aad6687e-07bc-45ab-a1fd-f5440229b528",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:35.304000-08:00",
    "startTime": "2025-02-10T20:18:35.615000-08:00",
    "endTime": "2025-02-10T20:18:35.651000-08:00",
    "status": "FAILED",
    "command": "fail command",
    "type": "SHELL",
    "exitCode": "127",
    "standardErrContent": "/codebuild/output/tmp/script.sh: 4: fail: not
found\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
```

```

        "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
        "encryptionDisabled": false
    }
},
{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",
    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
        "groupName": "group",
        "streamName": "stream",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
            "encryptionDisabled": false
        }
    }
}
]

```

```
}
```

Listar caixas de areia ()AWS CLI

CLI command

```
aws codebuild list-sandboxes --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --  
sort-order $SORT_ORDER
```

Sample request

```
aws codebuild list-sandboxes
```

Sample response

```
{  
  "ids": [  
    "s3-log-project-integ-test-temp173925062814985d64e0f-7880-41df-9a3c-  
fb6597a266d2:827a5243-0841-4b69-a720-4438796f6967",  
    "s3-log-project-integ-test-temp1739249999716bbd438dd-8bb8-47bd-  
ba6b-0133ac65b3d3:e2fa4eab-73af-42e3-8903-92fddaf9f378",  
    "s3-log-project-integ-test-  
temp17392474779450fbdacc2-2d6e-4190-9ad5-28f891bb7415:cd71e456-2a4c-4db4-ada5-  
da892b0bba05",  
    "s3-log-project-integ-test-temp17392246284164301421c-5030-4fa1-b4d3-  
ca15e44771c5:9e26ab3f-65e4-4896-a19c-56b1a95e630a",  
    "s3-log-project-integ-test-temp173921367319497056d8d-6d8e-4f5a-a37c-  
a62f5686731f:22d91b06-df1e-4e9c-a664-c0abb8d5920b",  
    "s3-log-project-integ-test-temp1739213439503f6283f19-390c-4dc8-95a9-  
c8480113384a:82cc413e-fc46-47ab-898f-ae23c83a613f",  
    "s3-log-project-integ-test-temp1739054385570b1f1ddc2-0a23-4062-  
bd0c-24e9e4a99b99:c02562f3-2396-42ec-98da-38e3fe5da13a",  
    "s3-log-project-integ-test-temp173905400540237dab1ac-1fde-4dfb-a8f5-  
c0114333dc89:d2f30493-f65e-4fa0-a7b6-08a5e77497b9",  
    "s3-log-project-integ-test-  
temp17390534055719c534090-7bc4-48f1-92c5-34acaec5bf1e:df5f1c8a-f017-43b7-91ba-  
ad2619e2c059",  
    "s3-log-project-integ-test-temp1739052719086a61813cc-  
ebb9-4db4-9391-7f43cc984ee4:d61917ec-8037-4647-8d52-060349272c4a",  
    "s3-log-project-integ-test-temp173898670094078b67edb-  
c42f-42ed-9db2-4b5c1a5fc66a:ce33dfbc-beeb-4466-8c99-a3734a0392c7",  
  ]  
}
```

```
"s3-log-project-integ-test-  
temp17389863425584d21b7cd-32e2-4f11-9175-72c89ecaffef:046dadf0-1f3a-4d51-a2c0-  
e88361924acf",  
  "s3-log-project-integ-test-  
temp1738985884273977ccd23-394b-46cc-90d3-7ab94cf764dc:0370dc41-9339-4b0a-91ed-51929761b244",  
  "s3-log-project-integ-test-temp1738985365972241b614f-8e41-4387-  
bd25-2b8351fbc9e0:076c392a-9630-47d8-85a9-116aa34edfff",  
  "s3-log-project-integ-test-  
temp1738985043988a51a9e2b-09d6-4d24-9c3c-1e6e21ac9fa8:6ea3949c-435b-4177-  
aa4d-614d5956244c",  
  "s3-log-project-integ-test-temp1738984123354c68b31ad-49d1-4f4b-981d-  
b66c00565ff6:6c3fff6c-815b-48b5-ada3-737400a6dee8",  
  "s3-log-project-integ-test-  
temp1738977263715d4d5bf6c-370a-48bf-8ea6-905358a6cf92:968a0f54-724a-42d1-9207-6ed854b2fae8",  
  "s3-log-project-integ-test-  
temp173897358796816ce8d7d-2a5e-41ef-855b-4a94a8d2795d:80f9a7ce-930a-402e-934e-  
d8b511d68b04",  
  "s3-log-project-integ-test-temp17389730633301af5e452-0966-467c-  
b684-4e36d47f568c:cabbe989-2e8a-473c-af25-32edc8c28646",  
  "s3-log-project-integ-test-temp1738901503813173fd468-  
b723-4d7b-9f9f-82e88d17f264:f7126a4a-b0d5-452f-814c-fea73718f805",  
  "s3-log-project-integ-test-temp1738890502472c13616fb-  
bd0f-4253-86cc-28b74c97a0ba:c6f197e5-3a53-45b6-863e-0e6353375437",  
  "s3-log-project-integ-test-  
temp17388903044683610daf3-8da7-43c6-8580-9978432432ce:d20aa317-8838-4966-  
bbfc-85b908213df1",  
  "s3-log-project-integ-test-temp173888857196780b5ab8b-e54b-44fd-a222-  
c5a374fffe96:ab4b9970-ffae-47a0-b3a8-7b6790008cad",  
  "s3-log-project-integ-test-temp1738888336931c11d378d-e74d-49a4-  
a723-3b92e6f7daac:4922f0e8-9b7d-4119-9c9f-115cd85e703e",  
  "s3-log-project-integ-test-temp17388881717651612a397-c23f-4d88-  
ba87-2773cd3fc0c9:be91c3fc-418e-4feb-8a3a-ba58ff8f4e8a",  
  "s3-log-project-integ-test-  
temp17388879727174c3c62ed-6195-4afb-8a03-59674d0e1187:a48826a8-3c0d-43c5-  
a1b5-1c98a0f978e9",  
  "s3-log-project-integ-test-temp1738885948597cef305e4-b8b4-46b0-a65b-  
e2d0a7b83294:c050e77d-e3f8-4829-9a60-46149628fe96",  
  "s3-log-project-integ-test-temp173888561463001a7d2a8-  
e4e4-4434-94db-09d3da9a9e17:8c3ac3f5-7111-4297-aec9-2470d3ead873",  
  "s3-log-project-integ-test-  
temp1738869855076eb19cafd-04fe-41bd-8aa0-40826d0c0d27:d25be134-05cb-404a-85da-  
ac5f85d2d72c",  
  "s3-project-integ-test-temp1738868157467148eacfc-d39b-49fc-a137-  
e55381cd2978:4909557b-c221-4814-b4b6-7d9e93d37c35",
```

```

        "s3-project-integ-test-temp1738820926895abec0af2-
e33d-473c-9cf4-2122dd9d6876:8f5cf218-71d6-40a4-a4be-6cacebd7765f",
        "s3-project-integ-test-temp173881998877574f969a6-1c2e-4441-b463-
ab175b45ce32:04396851-c901-4986-9117-585528e3877f",
        "s3-project-integ-test-temp17388189812309abd2604-29ba-4cf6-
b6bf-073207b7db9c:540075c7-f5ec-41e8-9341-2233c09247eb",
        "s3-project-integ-test-temp1738818843474d3ea9ac1-b609-461b-
bbdb-2da245c9bc96:865d4c3c-fbfe-4ece-9c92-d0c928341404",
        "s3-project-integ-test-temp1738818542236006e9169-e6d9-4344-9b59-
f557e7aec619:1f9ffa87-da15-4290-83e2-eebdd877497b",
        "s3-project-integ-test-
temp173881809557486ad11fd-7931-48d7-81d5-499cea52a6bc:c4c2efc4-685f-4e13-8b0f-1ef85ec300b1",
        "s3-project-integ-test-temp173881794103322941020-3f0b-49c3-b836-
fcd818ec9484:0344cfba-de48-456d-b2a8-6566bd4a5d6e",
        "s3-project-integ-test-temp1738817680747b93d0d0b-ea16-497f-9559-
af25ee6dcfdf:654a3a55-d92a-4dc6-8da8-56fd4d40d7e1",
        "s3-project-integ-test-temp17388174027191255c3da-086c-4270-b047-
acac0b7bee0d:b7e82740-2c69-42fc-ab5a-dbf15bc016a1",
        "s3-project-integ-test-temp1738817099799016e7fa3-b9b5-46a2-
bcd5-0888c646743f:8705a6a4-79ff-427a-a1c3-85c4e8fe462e",
        "s3-project-integ-test-temp1738816479281bb0c3606-5ebf-4623-
bed5-12b60e9d3512:f23fc74b-a981-4835-8e28-375fcd4c99e4",
        "s3-project-integ-test-
temp1738816263585c939a133-4d37-482c-9238-1dbff34b7674:ca28e234-0045-4ae6-8732-938b17597f50",
        "s3-project-integ-test-
temp173881580873072d18733-8fe4-43b1-83f7-95f25bb27ccf:c6f0f55b-5736-47c7-
a3aa-1b8461a6d5ed"
    ]
}

```

Tutorial: conectando-se a uma sandbox usando SSH

Este tutorial mostra como se conectar a uma CodeBuild sandbox usando um cliente SSH.

Para concluir este tutorial, você deve primeiramente:

- Certifique-se de ter um AWS CodeBuild projeto existente.
- Configure as permissões apropriadas do IAM configuradas para a função CodeBuild do seu projeto.
- Instale e configure AWS CLI em sua máquina local.

Etapa 1: iniciar uma sandbox

Para iniciar uma CodeBuild sandbox no console

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação. Escolha o projeto de compilação e, em seguida, escolha Debug build.

The screenshot shows the AWS CodeBuild console interface for a project named 'sandbox-project'. The breadcrumb trail at the top is 'Developer Tools > CodeBuild > Build projects > sandbox-project'. The page title is 'sandbox-project'. There are several action buttons: 'Actions' (dropdown), 'Create trigger', 'Edit', 'Clone', 'Debug build', 'Start build with overrides', and 'Start build' (highlighted in orange). Below the title is a 'Configuration' section with a table of settings:

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::012345678910:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

Below the configuration table is a tabbed interface with 'Project details' selected. Other tabs include 'Build history', 'Batch history', 'Build triggers', 'Metrics', and 'Debug sessions'. The 'Project configuration' section shows the project name 'sandbox-project', its ARN 'arn:aws:codebuild:us-east-1:012345678910:project/sandbox-project', and its description '-'. The 'Build badge' is shown as 'Disabled'. An 'Edit' button is located at the top right of this section.

3. Na guia Cliente SSH, escolha Iniciar sandbox.

The screenshot shows the 'Debug build' page in the AWS CodeBuild console. The breadcrumb trail is 'Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build'. The page title is 'Debug build'. There are three tabs: 'Run Command', 'SSH Client' (selected), and 'Session Manager'. Below the tabs is a light blue information box with an 'i' icon and the title 'Connect to your SSH client with sandbox'. It contains two bullet points: 'Launches a sandbox environment with SSH connectivity.' and 'Connect directly using SSH clients or your preferred IDE.' A 'Learn more' link with an external icon is at the top right of the box. At the bottom right of the page is an orange 'Start sandbox' button.

4. O processo de inicialização do sandbox pode levar algum tempo. Você pode se conectar à sandbox quando seu status mudar para RUN_SANDBOX.

Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build

Debug build

Run CommandSSH ClientSession Manager

Sandbox is running

Your sandbox `sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256` is ready and available for use.

Stop sandbox

TerminalVisual Studio CodeIntelliJ IDEA

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Make sure your CLI user has the `codebuild:StartSandboxConnection` permission. For more information, see [AWS CLI authentication](#) documentation.

Connect to your sandbox environment with following command:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:012345678910:sandbox/sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256
```

Sandbox phasesSandbox logsSandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
QUEUED	Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
PROVISIONING	Succeeded	-	4 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	Succeeded	-	6 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
RUN_SANDBOX	-	-	-	Apr 1, 2025 4:33 PM (UTC-7:00)	-

Etapa 2: Modificar a configuração SSH local

Se você estiver se conectando ao sandbox pela primeira vez, precisará realizar um processo de configuração único usando as seguintes etapas:

Para modificar a configuração SSH local no console

1. Localize os comandos de configuração do seu sistema operacional.

2. Abra seu terminal local e, em seguida, copie e execute os comandos fornecidos para baixar e executar o script para definir sua configuração SSH local. Por exemplo, se seu sistema operacional for macOS, use o seguinte comando:

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Depure compilações com sandbox CodeBuild

Versão da API 2016-10-06 685

3. O script de configuração adicionará as configurações necessárias para se conectar às suas sandboxes. Você será solicitado a aceitar essas alterações.
4. Após a configuração bem-sucedida, uma nova entrada de configuração SSH para CodeBuild sandbox será criada.

```
Host codebuild-sandbox-ssh*
  StrictHostKeyChecking no
  LogLevel INFO
  ForwardAgent yes
  ControlMaster auto
  ControlPersist 10m
  ProxyCommand ssh -C "/Users/[redacted]/.aws/codebuild-dev-env/codebuild-sandbox-connect.sh %n"
```

Etapa 3: conectar-se ao sandbox

Para modificar a configuração SSH local no console

1. Configure a AWS CLI autenticação e garanta que seu AWS CLI usuário tenha a `codebuild:StartSandboxConnection` permissão. Para obter mais informações, consulte [Autenticação usando credenciais de usuário do IAM AWS CLI no Guia do](#) usuário da interface de linha de AWS comando para a versão 1.
2. Conecte-se à sua sandbox com o seguinte comando:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>
```

Note

Para solucionar falhas de conexão, use o `-v` sinalizador para ativar a saída detalhada. Por exemplo, `ssh -v codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>`

Para obter orientações adicionais sobre solução de problemas, consulte [Solução de problemas de conexão SSH do AWS CodeBuild sandbox](#).

Etapa 4: analise seus resultados

Depois de conectado, você pode depurar falhas de compilação, testar comandos de compilação, experimentar alterações na configuração e verificar as variáveis e dependências do ambiente com sua sandbox.

Solução de problemas de conexão SSH do AWS CodeBuild sandbox

Use as informações deste tópico para ajudá-lo a identificar, diagnosticar e resolver problemas de conexão SSH do CodeBuild sandbox.

Tópicos

- [StartSandboxConnectionInvalidInputException erro quando o SSH entra no ambiente CodeBuild sandbox](#)
- [Erro: “Não é possível localizar as credenciais” quando o SSH entra no ambiente sandbox CodeBuild](#)
- [StartSandboxConnectionAccessDeniedException erro quando o SSH entra no ambiente CodeBuild sandbox](#)
- [Erro: “ssh: não foi possível resolver o nome do host” quando o SSH está no ambiente sandbox CodeBuild](#)

StartSandboxConnectionInvalidInputException erro quando o SSH entra no ambiente CodeBuild sandbox

Problema: ao tentar se conectar a um ambiente CodeBuild sandbox usando o comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, você pode encontrar um `InvalidInputException` erro como:

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection operation: Failed to start SSM session for {sandbox-arn}
User: arn:aws:sts::<account-ID>:assumed-role/<service-role-name>/AWSCodeBuild-<UUID>
is not authorized to perform: ssm:StartSession on resource.
```

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection operation: Failed to start SSM session for
sandbox <sandbox-arn>: codebuild:<UUID> is not connected.
```

Possível causa:

- Agente do Amazon EC2 Systems Manager ausente: a imagem de compilação não tem o agente SSM devidamente instalado ou configurado.
- Permissões insuficientes: a função de serviço do CodeBuild projeto não tem as permissões SSM necessárias.

Solução recomendada: se você estiver usando uma imagem personalizada para sua compilação, faça o seguinte.

1. Instalar o SSM Agent do . Para obter mais informações, consulte [Instalando e desinstalando manualmente o SSM Agent em EC2 instâncias da Amazon para Linux](#) no. A versão do SSM Agent deve ser 3.0.1295.0 ou posterior.
2. Copie o arquivo, <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/7.0/amazon-ssm-agent.json> para o diretório na sua imagem. /etc/amazon/ssm/ Isso ativa o Modo de Contêiner no agente SSM.
3. Certifique-se de que a função de serviço do seu CodeBuild projeto tenha as seguintes permissões e reinicie o ambiente sandbox:

```
{
  "Effect": "Allow",
  "Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
    "ssmmessages:OpenDataChannel"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:build/*",
    "arn:aws:ssm:region::document/AWS-StartSSHSession"
  ]
}
```

Erro: “Não é possível localizar as credenciais” quando o SSH entra no ambiente sandbox CodeBuild

Problema: ao tentar se conectar a um ambiente CodeBuild sandbox usando o comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, você pode encontrar o seguinte erro de credenciais:

```
Unable to locate credentials. You can configure credentials by running
```

```
"aws configure".
```

Possível causa: AWS as credenciais não foram configuradas corretamente em seu ambiente local.

Solução recomendada: configure suas AWS CLI credenciais seguindo a documentação oficial: [Definindo as configurações do Guia do usuário da AWS CLI](#) interface de linha de AWS comando para a versão 2.

StartSandboxConnectionAccessDeniedException erro quando o SSH entra no ambiente CodeBuild sandbox

Problema: ao tentar se conectar a um ambiente CodeBuild sandbox usando o comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, você pode encontrar o seguinte erro de permissão:

```
An error occurred (AccessDeniedException) when calling the StartSandboxConnection operation:
User: arn:aws:sts::account-id:assumed-role/role-name
is not authorized to perform: codebuild:StartSandboxConnection on resource:
sandbox-arn
because no identity-based policy allows the codebuild:StartSandboxConnection action
```

Causa possível: suas AWS credenciais não têm as CodeBuild permissões necessárias para realizar essa operação.

Solução recomendada: certifique-se de que o usuário ou a função do IAM associada às suas AWS CLI credenciais tenha as seguintes permissões:

```
{
  "Effect": "Allow",
  "Action": [
    "codebuild:StartSandboxConnection"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:sandbox/*"
  ]
}
```

Erro: “ssh: não foi possível resolver o nome do host” quando o SSH está no ambiente sandbox CodeBuild

Problema: ao tentar se conectar a um ambiente de CodeBuild sandbox usando o comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, você encontra o seguinte erro de resolução de nome de host:

```
ssh: Could not resolve hostname
```

Possível causa: Esse erro geralmente ocorre quando o script de conexão de CodeBuild sandbox necessário não foi executado corretamente em seu ambiente local.

A solução recomendada:

1. Baixe o script de conexão do CodeBuild sandbox.
2. Execute o script em seu terminal para estabelecer a configuração SSH necessária.
3. Tente novamente sua conexão SSH com o ambiente sandbox.

Depurar compilações com o Gerenciador de Sessões

Em AWS CodeBuild, você pode pausar uma compilação em execução e, em seguida, usar o Gerenciador de AWS Systems Manager Sessões para se conectar ao contêiner de compilação e visualizar o estado do contêiner.

Note

Esse atributo não está disponível em ambientes Windows.

Tópicos

- [Pré-requisitos](#)
- [Pausar a compilação](#)
- [Iniciar a compilação](#)
- [Conectar-se ao contêiner de compilação](#)
- [Retomar a compilação](#)

Pré-requisitos

Para permitir que o Gerenciador de Sessões seja usado com a sessão de compilação, é necessário habilitar a conexão de sessão para a compilação. Há dois pré-requisitos:

- CodeBuild As imagens com curadoria padrão do Linux já têm o agente SSM instalado e o agente SSM ativado. ContainerMode

Se você estiver usando uma imagem personalizada para a compilação, faça o seguinte:

1. Instalar o SSM Agent do . Para obter mais informações, consulte [Instalar manualmente o agente SSM em EC2 instâncias para Linux](#) no Guia do AWS Systems Manager usuário. A versão do Agente do SSM deve ser 3.0.1295.0 ou posterior.
2. Copie o arquivo <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent.json> para o diretório na sua imagem. /etc/amazon/ssm/ Isso habilita o modo de contêiner no agente do SSM.

Note

Imagens personalizadas exigiriam o agente do SSM mais atualizado para que esse atributo funcionasse conforme o esperado.

- A função CodeBuild de serviço deve ter a seguinte política de SSM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
    }
  ]
}
```

Você pode fazer com que o CodeBuild console anexe automaticamente essa política à sua função de serviço ao iniciar a compilação. Se preferir, você poderá anexar essa política ao perfil de serviço manualmente.

- Se você tiver a atividade de sessão de auditoria e registro ativada nas preferências do Systems Manager, a função CodeBuild de serviço também deverá ter permissões adicionais. As permissões são diferentes, dependendo do local onde os logs são armazenados.

CloudWatch Registros

Se estiver usando CloudWatch Logs para armazenar seus registros, adicione a seguinte permissão à função CodeBuild de serviço:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:<log-group-name>:*"
    }
  ]
}
```

Amazon S3

Se estiver usando o Amazon S3 para armazenar seus registros, adicione a seguinte permissão à função de CodeBuild serviço:

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

```
"Effect": "Allow",
"Action": [
  "s3:GetEncryptionConfiguration",
  "s3:PutObject"
],
"Resource": [
  "arn:aws:s3:::<bucket-name>",
  "arn:aws:s3:::<bucket-name>/*"
]
}
]
```

Para obter mais informações, consulte [Auditing and logging session activity](#) no Guia do usuário do AWS Systems Manager .

Parar a compilação

Para pausar a compilação, insira o comando `codebuild-breakpoint` em qualquer uma das fases de compilação no arquivo `buildspec`. A compilação será pausada nesse momento, o que permite a você se conectar ao contêiner de compilação e visualizar o contêiner no estado atual.

Por exemplo, adicione o seguinte às fases de compilação do arquivo `buildspec`.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Esse código cria o arquivo `/tmp/hello-world` e, depois, pausa a compilação nesse momento.

Iniciar a compilação

Para permitir que o Gerenciador de Sessões seja usado com a sessão de compilação, é necessário habilitar as conexões de sessão para a compilação. Para fazer isso, ao iniciar a compilação, siga as estas etapas:

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. No painel de navegação, selecione Projetos de compilação. Selecione o projeto de compilação e, depois, escolha Iniciar compilação com substituições.
3. Escolha Advanced build overrides (Substituições de compilação avançadas).
4. Na seção Ambiente, escolha a opção Habilitar conexão de sessão. Se essa opção não for selecionada, todos os comandos codebuild-breakpoint e codebuild-resume serão ignorados.
5. Faça todas as outras alterações desejadas e escolha Iniciar compilação.
6. Monitore o status da compilação no console. Quando a sessão está disponível, o link do Gerenciador de Sessões da AWS é exibido na seção Status da compilação.

Conectar-se ao contêiner de compilação

É possível se conectar ao contêiner de compilação de uma duas formas:

CodeBuild console

Em um navegador web, abra o link do Gerenciador de Sessões do AWS para se conectar ao contêiner de compilação. É aberta uma sessão de terminal que permite a você navegar e controlar o contêiner de compilação.

AWS CLI

Note

A máquina local deve ter o plug-in do Gerenciador de Sessões instalado para esse procedimento. Para obter mais informações, consulte [Instalar o plug-in do gerenciador de sessões para a AWS CLI](#) no Guia do AWS Systems Manager usuário.

1. Chame a API batch-get-builds com o ID da compilação para obter informações sobre a compilação, incluindo o identificador de destino da sessão. O nome da propriedade do identificador de destino da sessão varia de acordo com o tipo de saída do comando aws. É por isso que `--output json` é adicionado ao comando.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Copie o valor da propriedade `sessionTarget`. O nome da propriedade `sessionTarget` pode variar dependendo do tipo de saída do comando aws. É por isso que `--output json` foi adicionado ao comando na etapa anterior.

3. Use o comando a seguir para se conectar ao contêiner de compilação.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Neste exemplo, verifique se o arquivo `/tmp/hello-world` existe e contém o texto `Hello World`.

Retomar a compilação

Depois de terminar de examinar o contêiner de compilação, emita o comando `codebuild-resume` do shell de contêiner.

```
$ codebuild-resume
```

Excluir compilações em AWS CodeBuild

Você pode usar o AWS CLI ou o AWS SDKs para excluir compilações. AWS CodeBuild

Tópicos

- [Excluir compilações \(AWS CLI\)](#)
- [Excluir compilações \(AWS SDKs\)](#)

Excluir compilações (AWS CLI)

Execute o comando `batch-delete-builds`:

```
aws codebuild batch-delete-builds --ids ids
```

No comando anterior, substitua o seguinte espaço reservado:

- ***ids***: Cadeia de caracteres obrigatória. A IDs das compilações a serem excluídas. Para especificar várias compilações, separe o ID de cada compilação com um espaço. Para obter uma lista de compilações IDs, consulte os seguintes tópicos:
 - [Veja uma lista de build IDs \(AWS CLI\)](#)
 - [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#)

Em caso de êxito, uma matriz `buildsDeleted` será exibida na saída, contendo o Nome de recurso da Amazon (ARN) de cada compilação que foi excluída com êxito. Informações sobre compilações que não foram excluídas com êxito são exibidas na saída dentro de uma matriz `buildsNotDeleted`.

Por exemplo, se você executar este comando:

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Informações semelhantes às seguintes são exibidas na saída:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
  ]
}
```

Excluir compilações (AWS SDKs)

Para obter informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Repita as compilações manualmente em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para repetir manualmente uma única compilação ou uma compilação em AWS CodeBuild lote.

Tópicos

- [Repetir uma compilação manualmente \(console\)](#)
- [Repetir uma compilação manualmente \(AWS CLI\)](#)

- [Repetir uma compilação manualmente \(AWS SDKs\)](#)

Repetir uma compilação manualmente (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Execute um destes procedimentos:
 - Se a *build-ID* página *build-project-name*: for exibida, escolha Repetir compilação.
 - No painel de navegação, selecione Histórico de compilação. Na lista de compilações, selecione a caixa da compilação e depois selecione Interromper compilação.
 - No painel de navegação, selecione Projetos de compilação. Na lista de projetos de compilação, na coluna Nome, selecione o link para o nome do projeto de compilação. Na lista de compilações, selecione a caixa da compilação e depois selecione Interromper compilação.

Note

Por padrão, somente as 100 compilações ou projetos de compilação mais recentes são exibidos. Para visualizar mais compilações ou projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Compilações por página ou Projetos por página ou use as setas para voltar e avançar.

Repetir uma compilação manualmente (AWS CLI)

- Execute o comando `retry-build`:

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

No comando anterior, substitua o seguinte espaço reservado:

- *<build-id>*: Cadeia de caracteres obrigatória. O ID da compilação ou da compilação em lote a ser repetida. Para obter uma lista de compilações IDs, consulte os seguintes tópicos:
 - [Veja uma lista de build IDs \(AWS CLI\)](#)
 - [Exibir uma lista de compilações em lote IDs \(AWS CLI\)](#)
 - [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#)

- [Exibir uma lista de compilações em lote IDs para um projeto de compilação \(AWS CLI\)](#)
- `--idempotency-token`: opcional. Se você executar o comando `retry-build` com a opção, um identificador, ou token, exclusivo que diferencia maiúsculas e minúsculas será incluído na solicitação `retry-build`. O token é válido por 5 minutos após a solicitação. Se você repetir a `retry-build` solicitação com o mesmo token, mas alterar um parâmetro, CodeBuild retornará um erro de incompatibilidade de parâmetros.

Repetir uma compilação manualmente (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Repetir compilações automaticamente em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para repetir automaticamente suas compilações. AWS CodeBuild Com a repetição automática ativada, CodeBuild chamará automaticamente `RetryBuild` usando a função de serviço do projeto após uma falha na construção até um limite especificado. Por exemplo, se o limite de repetição automática estiver definido como dois, CodeBuild chamará a `RetryBuild` API para repetir automaticamente sua compilação por até duas vezes adicionais.

Note

CodeBuild não suporta repetição automática para. CodePipeline

Tópicos

- [Repetir uma compilação automaticamente \(console\)](#)
- [Repetir uma compilação automaticamente \(AWS CLI\)](#)
- [Repetir automaticamente um build \(AWS SDKs\)](#)

Repetir uma compilação automaticamente (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Escolha Criar projeto. Para obter informações, consulte [Criar um projeto de compilação \(console\)](#) e [Executar uma compilação \(console\)](#).
 - Em Ambiente:
 - Em Limite de tentativas automáticas, insira o número máximo de repetições automáticas desejadas após a falha de uma compilação.
3. Em Ambiente, escolha Configuração adicional.
4. Continue com os valores padrão e escolha Criar projeto de compilação.

Repetir uma compilação automaticamente (AWS CLI)

- Execute o comando create-project:

```
aws codebuild create-project \
  --name "<project-name>" \
  --auto-retry-limit <auto-retry-limit> \
  --source "<source>" \
  --artifacts {<artifacts>} \
  --environment "{\"type\": \"<environment-type>\", \"image\": \"<image-type>\", \"computeType\": \"<compute-type>\"}" \
  --service-role "<service-role>"
```

No comando anterior, substitua os seguintes espaços reservados:

- *<auto-retry-limit>*: defina o limite de repetições automáticas para o número máximo de tentativas automáticas desejadas após uma compilação com falha.
- *<project-name>*, *<source>*, *<artifacts>*, *<environment-type>*, *<image-type>*, *<compute-type>*, e *<service-role>*: defina as configurações de projeto desejadas.

Repetir automaticamente um build (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Pare de incorporar AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para interromper uma compilação AWS CodeBuild.

Tópicos

- [Parar uma compilação \(console\)](#)
- [Parar uma compilação \(AWS CLI\)](#)
- [Parar uma compilação \(AWS SDKs\)](#)

Parar uma compilação (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Execute um destes procedimentos:
 - Se a **build-ID** página **build-project-name**: for exibida, escolha Parar construção.
 - No painel de navegação, selecione Histórico de compilação. Na lista de compilações, selecione a caixa da compilação, depois selecione Interromper compilação.
 - No painel de navegação, selecione Projetos de compilação. Na lista de projetos de compilação, na coluna Nome, selecione o link para o nome do projeto de compilação. Na lista de compilações, selecione a caixa da compilação, depois selecione Interromper compilação.

Note

Por padrão, somente as 100 compilações ou projetos de compilação mais recentes são exibidos. Para visualizar mais compilações ou projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Compilações por página ou Projetos por página ou use as setas para voltar e avançar.

Se AWS CodeBuild não for possível interromper uma compilação com êxito (por exemplo, se o processo de compilação já estiver concluído), o botão Parar será desativado ou talvez não apareça.

Parar uma compilação (AWS CLI)

- Execute o comando stop-build:

```
aws codebuild stop-build --id id
```

No comando anterior, substitua o seguinte espaço reservado:

- *id*: Cadeia de caracteres obrigatória. A ID de build a ser parada. Para obter uma lista de compilações IDs, consulte os seguintes tópicos:
 - [Veja uma lista de build IDs \(AWS CLI\)](#)
 - [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#)

Se a compilação for interrompida AWS CodeBuild com êxito, o buildStatus valor no build objeto na saída será STOPPED.

Se CodeBuild não for possível interromper a compilação com êxito (por exemplo, se a compilação já estiver concluída), o buildStatus valor no build objeto na saída será o status final da compilação (por exemplo, SUCCEEDED).

Parar uma compilação (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Interrompa a criação de lotes AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para interromper a criação de um lote AWS CodeBuild.

Note

Se você usa a computação Lambda em sua compilação em lote, a compilação em andamento do Lambda não pode ser interrompida.

Tópicos

- [Interromper uma compilação \(console\)](#)
- [Interromper uma compilação em lote \(AWS CLI\)](#)
- [Interromper uma compilação em lote \(AWS SDKs\)](#)

Interromper uma compilação (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Execute um destes procedimentos:
 - Se a **build-ID** página **build-project-name**: for exibida, escolha Parar construção.
 - No painel de navegação, selecione Histórico de compilação. Na lista de compilações, selecione a caixa da compilação, depois selecione Interromper compilação.
 - No painel de navegação, selecione Projetos de compilação. Na lista de projetos de compilação, na coluna Nome, selecione o link para o nome do projeto de compilação. Na lista de compilações, selecione a caixa da compilação, depois selecione Interromper compilação.

Note

Por padrão, somente as 100 compilações ou projetos de compilação mais recentes são exibidos. Para visualizar mais compilações ou projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Compilações por página ou Projetos por página ou use as setas para voltar e avançar.

Interromper uma compilação em lote (AWS CLI)

- Execute o comando [stop-build-batch](#):

```
aws codebuild stop-build-batch --id <batch-build-id>
```

No comando anterior, substitua o seguinte espaço reservado:

- **<batch-build-id>**: Cadeia de caracteres obrigatória. O identificador da compilação em lote a ser interrompida. Para obter uma lista de identificadores de compilação, consulte os seguintes tópicos:

- [Exibir uma lista de compilações em lote IDs \(AWS CLI\)](#)
- [Exibir uma lista de compilações em lote IDs para um projeto de compilação \(AWS CLI\)](#)

Interromper uma compilação em lote (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

O gatilho AWS CodeBuild é criado automaticamente

Você pode criar um trigger em um projeto para programar um build uma vez a cada hora, dia ou semana. Você também pode editar um gatilho para usar uma regra personalizada com uma expressão CloudWatch cron da Amazon. Por exemplo, usando uma expressão cron, você pode programar uma compilação em um horário específico todos os dias úteis. Para obter informações sobre como criar e editar gatilhos, consulte [Crie AWS CodeBuild gatilhos](#) e [Editar AWS CodeBuild gatilhos](#).

Tópicos

- [Crie AWS CodeBuild gatilhos](#)
- [Editar AWS CodeBuild gatilhos](#)

Crie AWS CodeBuild gatilhos

Você pode criar um trigger em um projeto para programar um build uma vez a cada hora, dia ou semana. Você também pode criar um gatilho usando uma regra personalizada com uma expressão CloudWatch cron da Amazon. Por exemplo, usando uma expressão cron, você pode programar uma compilação em um horário específico todos os dias úteis.

Note

Não é possível iniciar uma compilação em lote a partir de um gatilho de compilação, de um EventBridge evento da Amazon ou de uma AWS Step Functions tarefa.

Tópicos

- [Criar AWS CodeBuild gatilhos \(console\)](#)
- [Crie AWS CodeBuild gatilhos programaticamente](#)

Criar AWS CodeBuild gatilhos (console)

Use o procedimento a seguir para criar gatilhos usando o AWS Management Console.

Para criar um trigger

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Escolha o link para o projeto de build ao qual você deseja adicionar o trigger e, em seguida, escolha a guia Construir triggers.

Note

Por padrão, somente os cem projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Projetos por página ou use as setas para voltar e avançar.

4. Escolha Criar acionador.
5. Em Trigger name (Nome do gatilho), insira um nome.
6. Na lista suspensa Frequência, escolha a frequência do gatilho. Se você deseja criar uma frequência usando uma expressão cron, escolha Personalizado.
7. Especifique os parâmetros para a frequência do gatilho. Você pode digitar os primeiros caracteres de suas seleções na caixa de texto para filtrar os itens de menu suspenso.

Note

As horas e os minutos de início são baseados em zero. O minuto de início é um número entre 0 e 59. A hora de início é um número entre 0 e 23. Por exemplo, um gatilho diário que começa todos os dias às 12h15 tem uma hora de início de 12 e um minuto de início de 15. Um gatilho diário que começa todos os dias às 0h tem uma hora de início de 0 e um minuto de início de 0. Um gatilho diário que começa todos os dias às 23h59 tem uma hora de início de 23 e um minuto de início de 59.

Frequência	Parâmetros necessários	Detalhes
Por hora	Minuto de início	Use o menu suspenso Minuto inicial.
Por dia	Minuto de início Hora de início	Use o menu suspenso Minuto inicial. Use o menu suspenso Hora inicial.
Por semana	Minuto de início Hora de início Dia inicial	Use o menu suspenso Minuto inicial. Use o menu suspenso Hora inicial. Use o menu suspenso Dia inicial.

Frequência	Parâmetros necessários	Detalhes
Personalizado	Expressão cron	Digite uma expressão cron em Expressão cron. Uma expressão cron tem seis campos obrigatórios, separados por espaços em branco. Os campos especificam um valor de início para minuto, hora, dia do mês, mês, dia da semana e ano. Você pode usar curingas para especificar um intervalo, valores adicionais e mais. Por exemplo, a expressão cron 0 9 ? * MON-FRI * programa uma compilação todos os dias da semana às 9h. Para obter mais informações, consulte Expressões Cron no Guia do usuário do Amazon Events CloudWatch.

8. Selecione Habilitar este trigger.
9. (Opcional) Expanda Advanced section (Seção avançada). Em Source version (Versão da origem), digite a versão da sua origem.
 - Para o Amazon S3, digite o ID da versão que corresponde à versão do artefato de entrada que você deseja compilar. Se o campo Source version (Versão de origem) for deixado em branco, a versão mais recente será usada.
 - Para AWS CodeCommit, digite um ID de confirmação. Se o campo Source version (Versão de origem) for deixado em branco, o ID de confirmação HEAD da ramificação padrão será usado.
 - Para GitHub ou GitHub Enterprise, digite uma ID de confirmação, uma ID de pull request, um nome de ramificação ou um nome de tag que corresponda à versão do código-fonte que você deseja criar. Se você especificar um ID de solicitação, ele deve usar o formato `pr/pull-request-ID` (por exemplo, `pr/25`). Se você especificar um nome de ramificação, o ID de

confirmação HEAD da ramificação será usado. Se Source version estiver em branco, o ID de confirmação HEAD da ramificação padrão será usado.

- Para o Bitbucket, digite um ID de confirmação, um nome da ramificação ou um nome de tag que corresponda à versão do código-fonte que você deseja compilar. Se você especificar um nome de ramificação, o ID de confirmação HEAD da ramificação será usado. Se Source version estiver em branco, o ID de confirmação HEAD da ramificação padrão será usado.
10. (Opcional) Especifique um tempo limite entre 5 minutos e 2.160 minutos (36 horas). Esse valor especifica por quanto tempo uma compilação AWS CodeBuild tenta antes que ela pare. Se Hours (Horas) e Minutes (Minutos) forem deixados em branco, será usado o valor de tempo limite padrão especificado no projeto.
 11. Escolha Criar acionador.

Crie AWS CodeBuild gatilhos programaticamente

CodeBuild usa EventBridge as regras da Amazon para criar gatilhos. Você pode usar a EventBridge API para criar programaticamente acionadores de compilação para seus projetos. CodeBuild Consulte [Amazon EventBridge API Reference](#) para obter mais informações.

Editar AWS CodeBuild gatilhos

Você pode editar um trigger em um projeto para programar um build uma vez a cada hora, dia ou semana. Você também pode editar um gatilho para usar uma regra personalizada com uma expressão CloudWatch cron da Amazon. Por exemplo, usando uma expressão cron, você pode programar uma compilação em um horário específico todos os dias úteis. Para obter mais informações sobre como criar um trigger, consulte [Crie AWS CodeBuild gatilhos](#).

Tópicos

- [Editar AWS CodeBuild gatilhos \(console\)](#)
- [Edite AWS CodeBuild gatilhos programaticamente](#)

Editar AWS CodeBuild gatilhos (console)

Use o procedimento a seguir para editar gatilhos usando o AWS Management Console.

Como editar um trigger

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. No painel de navegação, selecione Projetos de compilação.
3. Selecione o link para o projeto de compilação que você deseja alterar e selecione Build triggers (Gatilhos da compilação).

 Note

Por padrão, somente os cem projetos de compilação mais recentes são exibidos. Para ver mais projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Projetos por página ou use as setas para voltar e avançar.

4. Escolha o botão de seleção próximo ao trigger a ser alterado e selecione Edit (Editar).
5. Na lista suspensa Frequência, escolha a frequência do gatilho. Se você deseja criar uma frequência usando uma expressão cron, escolha Personalizado.
6. Especifique os parâmetros para a frequência do gatilho. Você pode digitar os primeiros caracteres de suas seleções na caixa de texto para filtrar os itens de menu suspenso.

 Note

As horas e os minutos de início são baseados em zero. O minuto de início é um número entre 0 e 59. A hora de início é um número entre 0 e 23. Por exemplo, um gatilho diário que começa todos os dias às 12h15 tem uma hora de início de 12 e um minuto de início de 15. Um gatilho diário que começa todos os dias às 0h tem uma hora de início de 0 e um minuto de início de 0. Um gatilho diário que começa todos os dias às 23h59 tem uma hora de início de 23 e um minuto de início de 59.

Frequência	Parâmetros necessários	Detalhes
Por hora	Minuto de início	Use o menu suspenso Minuto inicial.
Por dia	Minuto de início	Use o menu suspenso Minuto inicial.
	Hora de início	Use o menu suspenso Hora inicial.

Frequência	Parâmetros necessários	Detalhes
Por semana	Minuto de início Hora de início Dia inicial	Use o menu suspenso Minuto inicial. Use o menu suspenso Hora inicial. Use o menu suspenso Dia inicial.
Personalizado	Expressão cron	Digite uma expressão cron em Expressão cron. Uma expressão cron tem seis campos obrigatórios, separados por espaços em branco. Os campos especificam um valor de início para minuto, hora, dia do mês, mês, dia da semana e ano. Você pode usar curingas para especificar um intervalo, valores adicionais e mais. Por exemplo, a expressão cron 0 9 ? * MON-FRI * programa uma compilação todos os dias da semana às 9h. Para obter mais informações, consulte Expressões Cron no Guia do usuário do Amazon Events. CloudWatch

7. Selecione Habilitar este trigger.

 Note

Você pode usar o CloudWatch console da Amazon em <https://console.aws.amazon.com/cloudwatch/> para editar a versão de origem, o tempo limite e outras opções que não estão disponíveis em AWS CodeBuild.

Edite AWS CodeBuild gatilhos programaticamente

CodeBuild usa EventBridge as regras da Amazon para criar gatilhos. Você pode usar a EventBridge API para editar programaticamente os acionadores de construção dos seus projetos. CodeBuild Consulte [Amazon EventBridge API Reference](#) para obter mais informações.

Veja os detalhes da construção em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para ver detalhes sobre compilações gerenciadas pelo CodeBuild.

Tópicos

- [Visualizar detalhes de compilação \(console\)](#)
- [Visualizar detalhes de compilação \(AWS CLI\)](#)
- [Visualizar detalhes de compilação \(AWS SDKs\)](#)
- [Transições de fase de compilação](#)

Visualizar detalhes de compilação (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Execute um destes procedimentos:
 - No painel de navegação, selecione Histórico de compilação. Na lista de compilações, na coluna Build run (Execução de compilação), selecione o link para a compilação.
 - No painel de navegação, selecione Projetos de compilação. Na lista de projetos de compilação, na coluna Name (Nome), selecione o link para o nome do projeto de compilação. Posteriormente, na lista de compilações, na coluna Build run (Execução de compilação), selecione o link para a compilação.

 Note

Por padrão, somente as 10 compilações ou projetos de compilação mais recentes são exibidos. Para visualizar mais compilações ou projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Compilações por página ou Projetos por página ou use as setas para voltar e avançar.

Visualizar detalhes de compilação (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Execute o comando batch-get-builds:

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

Substitua o seguinte espaço reservado:

- *ids*: Cadeia de caracteres obrigatória. Uma ou mais construções IDs para ver detalhes sobre. Para especificar mais de uma ID de build, separe-as com um espaço. Você pode especificar até 100 compilações IDs. Para obter uma lista de compilações IDs, consulte os seguintes tópicos:
 - [Veja uma lista de build IDs \(AWS CLI\)](#)
 - [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#)

Por exemplo, se você executar este comando:

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

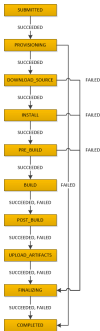
Se o comando for bem-sucedido, dados semelhantes aos descritos em [Como visualizar informações resumidas da compilação](#) serão exibidos na saída.

Visualizar detalhes de compilação (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Transições de fase de compilação

As compilações AWS CodeBuild prosseguem em fases:



 Important

A fase `UPLOAD_ARTIFACTS` é sempre tentada, mesmo se a fase `BUILD` falhar.

Veja uma lista de IDs incorporações AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para ver uma lista de compilações IDs para compilações gerenciadas por CodeBuild.

Tópicos

- Exibir uma lista de IDs compilações (console)
- Veja uma lista de build IDs (AWS CLI)
- Exibir uma lista de compilações em lote IDs (AWS CLI)
- Veja uma lista de build IDs (AWS SDKs)

Exibir uma lista de IDs compilações (console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. No painel de navegação, selecione Histórico de compilação.

 Note

Por padrão, apenas os 10 builds mais recentes são exibidos. Para visualizar mais builds, selecione o ícone de engrenagem e depois selecione um valor diferente para Builds per page (Builds por página) ou use as setas para voltar e avançar.

Veja uma lista de build IDs (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with CodeBuild, consulte [Referência da linha de comando](#) o.

- Execute o comando list-builds:

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

No comando anterior, substitua os seguintes espaços reservados:

- *sort-order*: string opcional usada para indicar como listar a compilação IDs. Os valores válidos são ASCENDING e DESCENDING.
- *next-token*: Cadeia de caracteres opcional. Se houver mais de 100 itens na lista em uma execução anterior, somente os primeiros 100 itens serão retornados, juntamente com uma string exclusiva de próximo token. Para obter o próximo lote de itens na lista, execute o comando novamente, adicionando o próximo token à chamada. Para obter todos os itens na lista, continue executando esse comando com cada próximo token subsequente, até que não haja mais tokens.

Por exemplo, se você executar este comando:

```
aws codebuild list-builds --sort-order ASCENDING
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
```

```
"codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
"codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
... The full list of build IDs has been omitted for brevity ...
"codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}
```

Se você executar este comando novamente:

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full
token has been omitted for brevity...MzY2OA==
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Exibir uma lista de compilações em lote IDs (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with CodeBuild, consulte [Referência da linha de comando](#) o.

- Execute o comando `list-build-batches`:

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

No comando anterior, substitua os seguintes espaços reservados:

- *sort-order*: string opcional usada para indicar como listar a compilação em lote IDs. Os valores válidos são ASCENDING e DESCENDING.
- *next-token*: Cadeia de caracteres opcional. Se houver mais de 100 itens na lista em uma execução anterior, somente os primeiros 100 itens serão retornados, juntamente com uma string exclusiva de próximo token. Para obter o próximo lote de itens na lista, execute o

comando novamente, adicionando o próximo token à chamada. Para obter todos os itens na lista, continue executando esse comando com cada próximo token subsequente, até que não haja mais tokens.

Por exemplo, se você executar este comando:

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Se você executar este comando novamente:

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Um resultado semelhante ao seguinte pode ser exibido na saída:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Veja uma lista de build IDs (AWS SDKs)

Para obter mais informações sobre como usar CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Veja uma lista de compilações IDs para um projeto de compilação em AWS CodeBuild

Você pode usar o AWS CodeBuild console, AWS CLI, ou AWS SDKs para ver uma lista de compilações IDs para um projeto de compilação em CodeBuild.

Tópicos

- [Exibir uma lista de compilações IDs para um projeto de compilação \(console\)](#)
- [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#)
- [Exibir uma lista de compilações em lote IDs para um projeto de compilação \(AWS CLI\)](#)
- [Exibir uma lista de IDs compilações para um projeto de construção \(AWS SDKs\)](#)

Exibir uma lista de compilações IDs para um projeto de compilação (console)

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. No painel de navegação, selecione Projetos de compilação. Na lista de projetos de build, na coluna Name (Nome), selecione o projeto de build.

Note

Por padrão, somente as 100 compilações ou projetos de compilação mais recentes são exibidos. Para visualizar mais compilações ou projetos de compilação, selecione o ícone de engrenagem e depois escolha outro valor para Compilações por página ou Projetos por página ou use as setas para voltar e avançar.

Exibir uma lista de IDs compilações para um projeto de construção (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Execute o comando `list-builds-for-project`, como se segue:

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

No comando anterior, substitua os seguintes espaços reservados:

- *project-name*: Cadeia de caracteres obrigatória usada para indicar o nome do projeto de construção para o qual listar as compilações IDs . Para obter uma lista de projetos de compilação, consulte [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#).
- *sort-order*: string opcional usada para indicar como listar a compilação IDs. Os valores válidos são ASCENDING e DESCENDING.
- *next-token*: Cadeia de caracteres opcional. Se houver mais de 100 itens na lista em uma execução anterior, somente os primeiros 100 itens serão retornados, juntamente com uma string exclusiva de próximo token. Para obter o próximo lote de itens na lista, execute o comando novamente, adicionando o próximo token à chamada. Para obter todos os itens na lista, continue executando esse comando com cada próximo token subsequente que é retornado, até que não haja mais tokens.

Por exemplo, se você executar este comando semelhante a este:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Um resultado como o seguinte pode aparecer na saída:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
  ]
}
```



```
"codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"  
]  
}
```

Se você executar este comando novamente:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --  
sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for  
brevity...MzY2OA==
```

Você pode ver um resultado como o seguinte na saída:

```
{  
  "ids": [  
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"  
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"  
    ... The full list of build IDs has been omitted for brevity ...  
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"  
  ]  
}
```

Exibir uma lista de compilações em lote IDs para um projeto de compilação (AWS CLI)

Para obter mais informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

Execute o comando `list-build-batches-for-project`, como se segue:

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-  
order sort-order --next-token next-token
```

No comando anterior, substitua os seguintes espaços reservados:

- *project-name*: Cadeia de caracteres obrigatória usada para indicar o nome do projeto de construção para o qual listar as compilações IDs . Para obter uma lista de projetos de compilação, consulte [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#).
- *sort-order*: string opcional usada para indicar como listar a compilação IDs. Os valores válidos são ASCENDING e DESCENDING.

- *next-token*: Cadeia de caracteres opcional. Se houver mais de 100 itens na lista em uma execução anterior, somente os primeiros 100 itens serão retornados, juntamente com uma string exclusiva de próximo token. Para obter o próximo lote de itens na lista, execute o comando novamente, adicionando o próximo token à chamada. Para obter todos os itens na lista, continue executando esse comando com cada próximo token subsequente que é retornado, até que não haja mais tokens.

Por exemplo, se você executar este comando semelhante a este:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Um resultado como o seguinte pode aparecer na saída:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Se você executar este comando novamente:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Você pode ver um resultado como o seguinte na saída:

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Exibir uma lista de IDs compilações para um projeto de construção (AWS SDKs)

Para obter mais informações sobre como usar AWS CodeBuild com o AWS SDKs, consulte [AWS SDKs e referência de ferramentas](#) o.

Relatórios de teste em AWS CodeBuild

Você pode criar relatórios CodeBuild que contenham detalhes sobre os testes que são executados durante as compilações. Você pode criar testes como testes de unidade, testes de configuração e testes funcionais.

Os seguintes formatos de arquivo de relatório de teste são compatíveis:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- Visual Studio TRX XML (.xml)

Note

A versão compatível mais recente do `cucumber-js` é 7.3.2.

Crie seus casos de teste com qualquer estrutura de teste que possa criar arquivos de relatório em um desses formatos (por exemplo, JUnit plug-in Surefire, TestNG ou Cucumber).

Para criar um relatório de teste, adicione um nome de grupo de relatórios ao arquivo `buildspec` de um projeto de compilação com informações sobre seus casos de teste. Quando você executa o projeto de compilação, os casos de teste são executados e um relatório de teste é criado. Um relatório de teste é criado no grupo de relatórios cada vez que os casos de teste são executados. Não é necessário criar um grupo de relatórios antes de executar os testes. Se você especificar um nome de grupo de relatórios, CodeBuild cria um grupo de relatórios para você ao executar seus relatórios. Se você quiser usar um grupo de relatórios que já existe, especifique seu ARN no arquivo `buildspec`.

Você pode usar um relatório de teste para ajudar a solucionar um problema durante uma execução de compilação. Se tiver muitos relatórios de teste de várias compilações de um projeto de compilação, você poderá usar seus relatórios de teste para visualizar tendências e taxas de teste e falha para ajudá-lo a otimizar compilações.

Um relatório expira 30 dias após sua criação. Não é possível visualizar um relatório de teste expirado. Se quiser manter relatórios de teste por mais de trinta dias, você poderá exportar os arquivos de dados brutos dos resultados do teste para um bucket do Amazon S3. Os arquivos de teste exportados não expiram. As informações sobre o bucket do S3 são especificadas ao criar o grupo de relatórios.

Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

Tópicos

- [Criar relatórios de teste](#)
- [Criar relatórios de cobertura de código](#)
- [Relatórios de descoberta automática em CodeBuild](#)
- [Grupos de relatórios](#)
- [Framework de teste](#)
- [Visualizar relatórios de teste](#)
- [Permissões de relatório de teste](#)
- [Status do relatório de teste](#)

Criar relatórios de teste

Para criar um relatório de teste, execute um projeto de compilação configurado com um a cinco grupos de relatórios em seu arquivo buildspec. Um relatório de teste é criado durante a execução. Ele contém os resultados dos casos de teste especificados para os grupos de relatórios. Um novo relatório de teste é gerado para cada compilação subsequente que usa o mesmo arquivo buildspec.

Para criar um relatório de teste

1. Crie um projeto de compilação. Para ter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#).
2. Configure o arquivo buildspec do seu projeto com informações do relatório de teste:

- a. Adicione uma seção `reports`: e especifique o ARN de um grupo de relatórios existente ou o nome de um grupo de relatórios.

Se você especificar um ARN, CodeBuild usa esse grupo de relatórios.

Se você especificar um nome, CodeBuild cria um grupo de relatórios para você usando o nome do seu projeto e o nome que você especificou, no formato `<project-name>-<report-group-name>`. Se o grupo de relatórios nomeado já existir, CodeBuild use esse grupo de relatórios.

- b. No grupo de relatórios, especifique o local dos arquivos que contêm os resultados do teste. Se você usar mais de um grupo de relatórios, especifique os locais do arquivo de resultados do teste para cada um. Um relatório de teste é criado sempre que o projeto de compilação é executado. Para obter mais informações, consulte [Especificar arquivos de teste](#).
- c. Na seção `commands` da sequência `post_build` ou `build`, especifique os comandos que executam os casos de teste especificados para os grupos de relatórios. Para obter mais informações, consulte [Especificar comandos de teste](#).

Veja um exemplo de seção `reports` do `buildspec`:

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Execute uma compilação do projeto de compilação. Para obter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#).
4. Quando a compilação estiver concluída, escolha a nova execução de compilação em Histórico de compilação na página do projeto. Escolha Reports (Relatórios) para visualizar o relatório de teste. Para obter mais informações, consulte [Visualizar relatórios de teste para uma compilação](#).

Criar relatórios de cobertura de código

CodeBuild permite gerar relatórios de cobertura de código para seus testes. Os seguintes relatórios de cobertura de código são fornecidos:

Cobertura de linha

A cobertura de linha mede quantas declarações os testes abrangem. Declaração é uma instrução única, que não inclui comentários nem condicionais.

```
line coverage = (total lines covered)/(total number of lines)
```

Cobertura de ramificação

A cobertura de ramificação mede quantas ramificações os testes abrangem de todas as ramificações possíveis de uma estrutura de controle, como uma declaração `if` ou `case`.

```
branch coverage = (total branches covered)/(total number of branches)
```

Os seguintes formatos de arquivo de relatório de cobertura de código são compatíveis:

- JaCoCo XML
- SimpleCov JSON¹
- Clover XML
- Cobertura XML
- INFORMAÇÕES SOBRE O LCOV

¹ [CodeBuild aceita relatórios de cobertura de código JSON gerados pelo simplecov, não pelo simplecov-json.](#)

Criar um relatório de cobertura de código

Para criar um relatório de cobertura de código, você executa um projeto de compilação configurado com pelo menos um grupo de relatórios de cobertura de código em seu arquivo `buildspec`. CodeBuild interpretará os resultados da cobertura do código e fornecerá um relatório da cobertura do código para a execução. Um novo relatório de teste é gerado para cada compilação subsequente que usa o mesmo arquivo `buildspec`.

Para criar um relatório de teste

1. Crie um projeto de compilação. Para ter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#).

2. Configure o arquivo `buildspec` do projeto com informações do relatório de teste:
 - a. Adicione uma `reports:` seção e especifique o nome do seu grupo de relatórios. CodeBuild cria um grupo de relatórios para você usando o nome do seu projeto e o nome que você especificou no formato `project-name-report-group-name-in-buildspec`. Se você já tiver um grupo de relatórios que deseja usar, especifique seu ARN. Se você usar o nome em vez do ARN, CodeBuild cria um novo grupo de relatórios. Para obter mais informações, consulte [Reports syntax in the buildspec file](#).
 - b. No grupo de relatórios, especifique o local dos arquivos que contêm os resultados da cobertura de código. Se você usar mais de um grupo de relatórios, especifique os locais do arquivo de resultados para cada grupo de relatórios. Um novo relatório de cobertura de código é criado sempre que o projeto de compilação é executado. Para obter mais informações, consulte [Especificar arquivos de teste](#).

Este é um exemplo que gera um relatório de cobertura de código para um arquivo de resultados JaCoCo XML localizado em `test-results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

- c. Na seção `commands` da sequência `build` ou `post_build`, especifique os comandos que executam a análise de cobertura de código. Para obter mais informações, consulte [Especificar comandos de teste](#).
3. Execute uma compilação do projeto de compilação. Para obter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#).
4. Quando a compilação estiver concluída, escolha a nova execução de compilação em Histórico de compilação na página do projeto. Escolha Relatórios para ver o relatório de cobertura de código. Para obter mais informações, consulte [Visualizar relatórios de teste para uma compilação](#).

Relatórios de descoberta automática em CodeBuild

Com a descoberta automática, CodeBuild pesquisa todos os seus arquivos de compilação após a conclusão da fase de compilação, pesquisa qualquer tipo de arquivo de relatório compatível e

cria automaticamente novos grupos e relatórios de relatórios de cobertura de teste e código. Para qualquer tipo de relatório descoberto, CodeBuild cria novos grupos de relatórios com o seguinte padrão:

```
<project-name>-<report-file-format>-AutoDiscovered
```

Note

Se os arquivos de relatório descobertos tiverem o mesmo tipo de formato, eles serão colocados no mesmo grupo de relatórios ou relatório.

A descoberta automática de relatórios é configurada pelas variáveis de ambiente do projeto:

CODEBUILD_CONFIG_AUTO_DISCOVER

Essa variável determina se a descoberta automática do relatório está desativada durante a compilação. Por padrão, a descoberta automática de relatórios está ativada para todas as compilações. Para desabilitar esse recurso, defina CODEBUILD_CONFIG_AUTO_DISCOVER como false.

CODEBUILD_CONFIG_AUTO_DISCOVER_DIR

(Opcional) Essa variável determina onde CodeBuild procura possíveis arquivos de relatório. Observe que, por padrão, CodeBuild pesquisa `**/*` por padrão.

Essas variáveis de ambiente podem ser modificadas durante a fase de compilação. Por exemplo, se você quiser ativar apenas a descoberta automática de relatórios para compilações na ramificação git main, você pode selecionar a ramificação git durante o processo de compilação e definir CODEBUILD_CONFIG_AUTO_DISCOVER como falso se a compilação não estiver na ramificação main. A descoberta automática de relatórios pode ser desativada usando o console ou usando variáveis de ambiente do projeto.

Tópicos

- [Configurar a descoberta automática de relatórios usando o console](#)
- [Configure a descoberta automática de relatórios usando variáveis de ambiente do projeto](#)

Configurar a descoberta automática de relatórios usando o console

Use o procedimento a seguir para configurar a descoberta automática de relatórios usando o console.

Para configurar a descoberta automática de relatórios usando o console

1. Crie um projeto de compilação ou escolha um projeto de compilação para editar. Para obter informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) ou [Alterar as configurações do projeto de construção em AWS CodeBuild](#).
2. Em Ambiente, selecione Configuração adicional.
3. Para desativar a descoberta automática de relatórios, em Descoberta automática de relatórios, selecione Desativar a descoberta automática de relatórios.
4. (Opcional) Em Diretório de descoberta automática - opcional, insira um padrão de diretório CodeBuild para pesquisar arquivos de formato de relatório compatíveis. Observe que a CodeBuild pesquisa é `**/*` feita por padrão.

Configure a descoberta automática de relatórios usando variáveis de ambiente do projeto

Use o procedimento a seguir para configurar a descoberta automática de relatórios usando as variáveis do ambiente do projeto.

Para configurar a descoberta automática de relatórios usando variáveis de ambiente do projeto

1. Crie um projeto de compilação ou escolha um projeto de compilação para editar. Para obter informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) ou [Alterar as configurações do projeto de construção em AWS CodeBuild](#).
2. Em Variáveis de ambiente, faça o seguinte:
 - a. Para desativar a descoberta automática de relatórios, em Nome, insira **CODEBUILD_CONFIG_AUTO_DISCOVER** e, em Valor, insira **false**. Isso desativa a descoberta automática de relatórios.
 - b. (Opcional) Em Nome, insira **CODEBUILD_CONFIG_AUTO_DISCOVER_DIR** e, em Valor, insira o diretório em que CodeBuild deve pesquisar os arquivos de formato de relatório compatíveis. Por exemplo, `output/*xml` pesquisa arquivos `.xml` no diretório `output`

Grupos de relatórios

Um grupo de relatórios contém relatórios de teste e especifica configurações compartilhadas. Use o arquivo buildspec para especificar os casos de teste a serem executados e os comandos para executá-los quando ele for compilado. Para cada grupo de relatórios configurado em um projeto de compilação, uma execução do projeto de compilação criará um relatório de teste. Várias execuções de um projeto de compilação configurado com um grupo de relatórios criam vários relatórios de teste nesse grupo de relatórios, cada um com resultados dos mesmos casos de teste especificados para esse grupo de relatórios.

Os casos de teste são especificados para um grupo de relatórios no arquivo buildspec de um projeto de compilação. É possível especificar até cinco grupos de relatórios em um projeto de compilação. Ao executar uma compilação, todos os casos de teste são executados. Um relatório de teste é criado com os resultados de cada caso de teste especificado para um grupo de relatórios. Cada vez que você executa uma nova compilação, os casos de teste são executados e um relatório de teste é criado com os novos resultados do teste.

Os grupos de relatórios podem ser usados em mais de um projeto de compilação. Todos os relatórios de teste criados com um grupo de relatórios compartilham a mesma configuração, como sua opção de exportação e permissões, mesmo que os relatórios de teste sejam criados usando projetos de compilação diferentes. Os relatórios de teste criados com um grupo de relatórios em vários projetos de compilação podem conter os resultados da execução de diferentes conjuntos de casos de teste (um conjunto de casos de teste para cada projeto de compilação). Isso ocorre porque você pode especificar diferentes arquivos de caso de teste para o grupo de relatórios no arquivo buildspec de cada projeto. Você também pode alterar os arquivos de caso de teste para um grupo de relatórios em um projeto de compilação editando seu arquivo buildspec. Execuções de compilação subsequentes criam relatórios de teste que contêm os resultados dos arquivos de caso de teste no buildspec atualizado.

Tópicos

- [Criar um grupo de relatórios](#)
- [Nomenclatura do grupo de relatórios](#)
- [Compartilhar grupos de relatórios](#)
- [Especificar arquivos de teste](#)
- [Especificar comandos de teste](#)
- [Marcar um grupo de relatórios em AWS CodeBuild](#)

- [Atualizar um grupo de relatórios](#)

Criar um grupo de relatórios

Você pode usar o CodeBuild console AWS CLI, o ou um arquivo buildspec para criar um grupo de relatórios. Sua função do IAM deve ter as permissões necessárias para criar um grupo de relatórios. Para obter mais informações, consulte [Permissões de relatório de teste](#).

Tópicos

- [Criar um grupo de relatórios \(buildspec\)](#)
- [Criar um grupo de relatórios \(console\)](#)
- [Criar um grupo de relatórios \(CLI\)](#)
- [Criar um grupo de relatórios \(AWS CloudFormation\)](#)

Criar um grupo de relatórios (buildspec)

Um grupo de relatórios criado com o uso do buildspec não exporta arquivos de resultado de teste bruto. Você pode visualizar seu grupo de relatórios e especificar as configurações de exportação. Para obter mais informações, consulte [Atualizar um grupo de relatórios](#).

Como criar um grupo de relatórios usando um arquivo buildspec

1. Escolha um nome de grupo de relatórios que não esteja associado a um grupo de relatórios na sua AWS conta.
2. Configure a seção `reports` do arquivo buildspec com esse nome. Neste exemplo, o nome do grupo de relatórios é `new-report-group` e os casos de teste de uso são criados com a JUnit estrutura:

```
reports:
  new-report-group: #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

O nome do grupo de relatórios também pode ser especificado usando variáveis de ambiente no buildspec:

```
version: 0.2
env:
  variables:
    REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
...
reports:
  $REPORT_GROUP_NAME:
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Para ter mais informações, consulte [Especificar arquivos de teste](#) e [Reports syntax in the buildspec file](#).

3. Na seção `commands`, especifique o comando para executar seus testes. Para obter mais informações, consulte [Especificar comandos de teste](#).
4. Execute o build. Quando a compilação estiver concluída, um grupo de relatórios será criado com um nome que usa o formato `project-name-report-group-name`. Para obter mais informações, consulte [Nomenclatura do grupo de relatórios](#).

Criar um grupo de relatórios (console)

O procedimento a seguir cria um grupo de relatórios usando a AWS Management Console.

Para criar um grupo de relatórios

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Grupos de relatórios.
3. Escolha Criar grupo de relatórios.
4. Em Report name group (Nome do grupo de relatórios), insira um nome para o grupo de relatórios.
5. (Opcional) Em Tags, insira o nome e o valor de todas as tags que você deseja que AWS os serviços de suporte usem. Use Adicionar linha para adicionar uma tag. É possível adicionar até 50 tags.

6. Se você quiser fazer upload dos dados brutos dos resultados do relatório de teste para um bucket do Amazon S3:
 - a. Selecione Exportar para o Amazon S3.
 - b. Em Nome do bucket do S3, insira o nome do bucket do S3.
 - c. (Opcional) Para Proprietário do bucket do S3, insira o identificador da conta da AWS que tem o bucket do S3. Isso permite que os dados do relatório sejam exportados para um bucket do Amazon S3 que pertence a uma conta diferente da conta que está executando a compilação.
 - d. Em Prefixo do caminho, insira o caminho no bucket do S3 onde deseja fazer upload dos resultados do teste.
 - e. Selecione Compactar dados do resultado do teste em um arquivo zip para compactar os arquivos de dados do resultado de teste bruto.
 - f. Expanda Configuração adicional para exibir opções de criptografia. Escolha uma das seguintes opções:
 - Chave AWS gerenciada padrão para usar a Chave gerenciada pela AWS para o Amazon S3. Para obter mais informações, consulte [Gerenciado pelo cliente CMKs](#) no Guia AWS Key Management Service do usuário. Esta é a opção de criptografia padrão.
 - Escolher uma chave personalizada para usar uma chave gerenciada pelo cliente criada e configurada por você. Para Chave de criptografia do AWS KMS, insira o ARN da chave de criptografia. O formato é `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>`. Para obter mais informações, consulte [Criar chaves do KMS](#) no Guia do usuário do AWS Key Management Service.
 - Desativar criptografia de artefatos para desabilitar a criptografia. Você pode escolher isso se quiser compartilhar seus resultados de teste ou publicá-los em um site estático. (Um site dinâmico pode executar códigos para descriptografar os resultados de teste.)

Para obter mais informações sobre criptografia de dados em repouso, consulte [Criptografia de dados](#).

 Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

7. Escolha Criar grupo de relatórios.

Criar um grupo de relatórios (CLI)

O procedimento a seguir cria um grupo de relatórios usando a AWS CLI.

Para criar um grupo de relatórios

1. Crie um arquivo chamado `CreateReportGroup.json`.
2. Dependendo de suas necessidades, copie um dos seguintes trechos de código JSON para `CreateReportGroup.json`:
 - Use o seguinte JSON para especificar que o grupo de relatórios de teste exporta arquivos de resultado de teste bruto para um bucket do Amazon S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    },
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```

```
}  
}
```

- *<bucket-name>* Substitua pelo nome do bucket do Amazon S3 e *<path>* pelo caminho no bucket para onde você deseja exportar os arquivos.
- Se desejar compactar os arquivos exportados, para `packaging`, especifique ZIP. Caso contrário, especifique NONE.
- `bucketOwner` é opcional e só será necessário se o bucket do Amazon S3 pertencer a uma conta diferente da conta que está executando a compilação.
- Use `encryptionDisabled` para especificar se deseja criptografar os arquivos exportados. Se você criptografar os arquivos exportados, insira a chave gerenciada pelo cliente. Para obter mais informações, consulte [Atualizar um grupo de relatórios](#).
- Use o seguinte JSON para especificar que seu relatório de teste não exporta arquivos de teste brutos:

```
{  
  "name": "<report-name>",  
  "type": "TEST",  
  "exportConfig": {  
    "exportConfigType": "NO_EXPORT"  
  }  
}
```

Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

3. Execute o seguinte comando:

```
aws codebuild create-report-group --cli-input-json file://  
CreateReportGroupInput.json
```


Criar um grupo de relatórios (AWS CloudFormation)

Utilize as instruções a seguir para criar um grupo de relatórios utilizando o modelo do AWS CloudFormation

Para criar um grupo de relatórios usando o modelo do AWS CloudFormation

Você pode usar um arquivo AWS CloudFormation de modelo para criar e provisionar um grupo de relatórios. Para obter mais informações, consulte o [Guia do usuário do AWS CloudFormation](#).

O modelo AWS CloudFormation YAML a seguir cria um grupo de relatórios que não exporta arquivos brutos de resultados de testes.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: NO_EXPORT
```

O modelo AWS CloudFormation YAML a seguir cria um grupo de relatórios que exporta arquivos brutos de resultados de testes para um bucket do Amazon S3.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: S3
        S3Destination:
          Bucket: amzn-s3-demo-bucket
          Path: path-to-folder-for-exported-files
          Packaging: ZIP
          EncryptionKey: my-KMS-encryption-key
          EncryptionDisabled: false
```

 Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

Nomenclatura do grupo de relatórios

Ao usar o console AWS CLI ou o AWS CodeBuild console para criar um grupo de relatórios, você especifica um nome para o grupo de relatórios. Se você usar a buildspec para criar um grupo de relatórios, ele será nomeado usando o formato *project-name-report-group-name-specified-in-buildspec*. Todos os relatórios criados pela execução de compilações desse projeto de compilação pertencem ao novo grupo de relatórios com o novo nome.

Se você não quiser CodeBuild criar um novo grupo de relatórios, especifique o ARN do grupo de relatórios no arquivo buildspec de um projeto de compilação. Você pode especificar o ARN de um grupo de relatórios em vários projetos de compilação. Depois que cada projeto de compilação é executado, o grupo de relatórios contém relatórios de teste criados por cada projeto de compilação.

Por exemplo, se você criar um grupo de relatórios com o nome `my-report-group` e usar seu nome em dois projetos de compilação diferentes nomeados `my-project-1` e `my-project-2` e criar uma compilação dos dois projetos, dois grupos de relatórios serão criados. O resultado são três grupos de relatórios com os seguintes nomes:

- `my-report-group`: não tem nenhum relatório de teste.
- `my-project-1-my-report-group`: contém relatórios com resultados de testes executados pelo projeto de compilação chamado `my-project-1`.
- `my-project-2-my-report-group`: contém relatórios com resultados de testes executados pelo projeto de compilação chamado `my-project-2`.

Se você usar o ARN do grupo de relatórios chamado `my-report-group` nos dois projetos e executar compilações de cada projeto, você ainda terá um grupo de relatórios (`my-report-group`). Esse grupo de relatórios contém relatórios de teste com resultados de testes executados pelos dois projetos de compilação.

Se você escolher um nome de grupo de relatórios que não pertença a um grupo de relatórios na sua conta da AWS, usar esse nome para um grupo de relatórios em um arquivo de compilação e executar uma compilação de seu projeto de compilação, um grupo de relatórios será criado. O

formato do nome do novo grupo de relatórios é *project-name-new-group-name*. Por exemplo, se não houver um grupo de relatórios em sua AWS conta com o nome new-report-group e especificá-lo em um projeto de compilação chamado test-project, uma execução de compilação criará um novo grupo de relatórios com o nome test-project-new-report-group.

Compartilhar grupos de relatórios

O compartilhamento do grupo de relatórios permite que várias AWS contas ou usuários visualizem um grupo de relatórios, seus relatórios não expirados e os resultados dos testes de seus relatórios. Neste modelo, a conta que possui o grupo de relatórios (proprietário) compartilha um grupo de relatórios com outras contas (consumidores). Um consumidor não pode editar um grupo de relatórios. Um relatório expira 30 dias após sua criação.

Tópicos

- [Compartilhar um grupo de relatórios](#)
- [Serviços relacionados](#)
- [Acessar grupos de relatórios compartilhados com você](#)
- [Cancelar o compartilhamento de um grupo de relatórios compartilhado](#)
- [Identificar um grupo de relatórios compartilhados](#)
- [Permissões do grupo de relatórios compartilhados](#)

Compartilhar um grupo de relatórios

Quando você compartilha um grupo de relatórios, o consumidor recebe acesso somente leitura ao grupo de relatórios e seus relatórios. O consumidor pode usar o AWS CLI para visualizar o grupo de relatórios, seus relatórios e os resultados do caso de teste de cada relatório. O consumidor não pode:

- Visualize um grupo de relatórios compartilhado ou seus relatórios no CodeBuild console.
- Editar um grupo de relatórios compartilhados.
- Usar o ARN do grupo de relatórios compartilhados em um projeto para executar um relatório. Uma compilação de projeto que especifique um grupo de relatórios compartilhados falhará.

Você pode usar o CodeBuild console para adicionar um grupo de relatórios a um compartilhamento de recursos existente. Se quiser adicionar o grupo de relatórios a um novo compartilhamento de recursos, primeiro você deve criá-lo no [console do AWS IAM](#).

Para compartilhar um grupo de relatórios com unidades organizacionais ou toda uma organização, é necessário habilitar o compartilhamento com AWS Organizations. Para obter mais informações, consulte [Habilitar o compartilhamento com o AWS Organizations](#) no Guia do usuário do AWS RAM .

Você pode usar o CodeBuild console, o AWS RAM console ou AWS CLI compartilhar grupos de relatórios que você possui.

Pré-requisito

Para compartilhar um grupo de relatórios, sua AWS conta precisa ser proprietária dele. Não é possível compartilhar um grupo de relatórios que tenha sido compartilhado com você.

Para compartilhar um grupo de relatórios que você possui (CodeBuild console)

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Grupos de relatórios.
3. Escolha o projeto que deseja compartilhar e selecione Compartilhar. Para obter mais informações, consulte [Create a resource share](#) no Guia do usuário do AWS RAM .

Para compartilhar grupos de relatórios que você possui (AWS RAM console)

Consulte [Creating a resource share](#) no Guia do usuário do AWS RAM .

Para compartilhar grupos de relatórios que você possui (AWS RAM comando)

Use o comando [create-resource-share](#).

Para compartilhar um grupo de relatórios que você possui (CodeBuild comando)

Use o comando [put-resource-policy](#):

1. Crie um arquivo chamado `policy.json` e copie o seguinte nele.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "consumer-aws-account-id-or-user"
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
```

```

    "codebuild:BatchGetReports",
    "codebuild:ListReportsForReportGroup",
    "codebuild:DescribeTestCases"],
    "Resource": "arn-of-report-group-to-share"
  }]
}

```

2. Atualize `policy.json` com o ARN do grupo de relatórios e os identificadores com quais compartilhá-lo. O exemplo a seguir concede acesso somente de leitura ao grupo de relatórios com o ARN para Alice e `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` para o usuário raiz da conta identificada por 123456789012. AWS

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::123456789012:user/Alice",
        "123456789012"
      ]
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"],
    "Resource": "arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group"
  }]
}

```

3. Execute o seguinte comando.

```

aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://policy.json

```

Serviços relacionados

O compartilhamento de grupos de relatórios se integra ao AWS Resource Access Manager (AWS RAM), um serviço que possibilita que você compartilhe seus AWS recursos com qualquer

AWS conta ou por meio AWS Organizations de. Com AWS RAM, você compartilha recursos de sua propriedade criando um compartilhamento de recursos que especifica os recursos e os consumidores com os quais compartilhá-los. Os consumidores podem ser AWS contas individuais, unidades organizacionais ou uma organização inteira em AWS Organizations. AWS Organizations

Para obter mais informações, consulte o Guia do usuário do [AWS RAM](#).

Acessar grupos de relatórios compartilhados com você

Para acessar um grupo de relatórios compartilhados, a função do IAM de um consumidor requer a permissão `BatchGetReportGroups`. É possível anexar a seguinte política ao perfil do IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetReportGroups"
  ]
}
```

Para obter mais informações, consulte [Usando políticas baseadas em identidade para AWS CodeBuild](#).

Cancelar o compartilhamento de um grupo de relatórios compartilhado

Um grupo de relatórios cancelado, incluindo seus relatórios e seus resultados de caso de teste, só pode ser acessado pelo proprietário. Se você cancelar o compartilhamento de um grupo de relatórios, qualquer AWS conta ou usuário com quem você o compartilhou anteriormente não poderá acessar o grupo de relatórios, seus relatórios ou os resultados dos casos de teste nos relatórios.

Para cancelar o compartilhamento de um grupo de relatórios de sua propriedade, é necessário removê-lo do compartilhamento de recursos. Você pode usar o AWS RAM console ou AWS CLI fazer isso.

Para cancelar o compartilhamento de um grupo de relatórios compartilhado que você possui (AWS RAM console)

Consulte [Atualização de um compartilhamento de recursos](#) no Guia do usuário do AWS RAM .

Para cancelar o compartilhamento de um grupo de relatórios compartilhado de sua propriedade (AWS RAM comando)

Use o comando [disassociate-resource-share](#).

Para cancelar o compartilhamento do grupo de relatórios que você possui (CodeBuild comando)

Execute o [delete-resource-policy](#) comando e especifique o ARN do grupo de relatórios que você deseja cancelar o compartilhamento:

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Identificar um grupo de relatórios compartilhados

Proprietários e consumidores podem usar o AWS CLI para identificar grupos de relatórios compartilhados.

Para identificar e obter informações sobre um grupo de relatórios compartilhados e seus relatórios, use os seguintes comandos:

- Para ver os grupos ARNs de relatórios compartilhados com você, execute [list-shared-report-groups](#):

```
aws codebuild list-shared-report-groups
```

- Para ver os relatórios em um grupo ARNs de relatórios, execute [list-reports-for-report-group](#) usando o ARN do grupo de relatórios:

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Para visualizar informações sobre casos de teste em um relatório, execute [describe-test-cases](#) usando o ARN do relatório:

```
aws codebuild describe-test-cases --report-arn report-arn
```

A saída será exibida como a seguir:

```
{
  "testCases": [
    {
      "status": "FAILED",
```

```
        "name": "Test case 1",
        "expired": 1575916770.0,
        "reportArn": "report-arn",
        "prefix": "Cucumber tests for agent",
        "message": "A test message",
        "durationInNanoSeconds": 1540540,
        "testRawDataPath": "path-to-output-report-files"
    },
    {
        "status": "SUCCEEDED",
        "name": "Test case 2",
        "expired": 1575916770.0,
        "reportArn": "report-arn",
        "prefix": "Cucumber tests for agent",
        "message": "A test message",
        "durationInNanoSeconds": 1540540,
        "testRawDataPath": "path-to-output-report-files"
    }
]
```

Permissões do grupo de relatórios compartilhados

Permissões para proprietários

Um proprietário do grupo de relatórios pode editar o grupo de relatórios e especificá-lo em um projeto para executar relatórios.

Permissões para clientes

Um consumidor do grupo de relatórios pode visualizar um grupo de relatórios, seus relatórios e os resultados do caso de teste para os relatórios. Um consumidor não pode editar um grupo de relatórios, ou seus relatórios, e não pode usá-lo para criar relatórios.

Especificar arquivos de teste

Especifique os arquivos de resultado do teste e seu local para cada grupo de relatórios na seção `reports` do arquivo `buildspec` do seu projeto de compilação. Para obter mais informações, consulte [Reports syntax in the buildspec file](#).

Veja a seguir uma seção de exemplo `reports` que especifica dois grupos de relatórios para um projeto de compilação. Um é especificado pelo ARN, o outro pelo nome. A seção `files` especifica

os arquivos que contêm os resultados do caso de teste. A seção `base-directory` opcional especifica o diretório onde os arquivos de caso de teste estão localizados. A seção `discard-paths` opcional especifica se os caminhos para testar arquivos de resultado enviados a um bucket do Amazon S3 serão descartados.

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  #surefire junit reports
  files:
    - '**/*'
  base-directory: 'surefire/target/surefire-reports'
  discard-paths: false

sampleReportGroup: #Cucumber reports from json plugin
  files:
    - 'cucumber-json/target/cucumber-json-report.json'
  file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Especificar comandos de teste

Especifique os comandos que executam seus casos de teste na seção `commands` do arquivo `buildspec`. Esses comandos executam os casos de teste especificados para seus grupos de relatórios na seção `reports` do arquivo `buildspec`. Veja a seguir uma seção `commands` de exemplo que inclui comandos para executar os testes em arquivos de teste:

```
commands:
  - echo Running tests for surefire junit
  - mvn test -f surefire/pom.xml -fn
  - echo
  - echo Running tests for cucumber with json plugin
  - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
    cucumber-json/pom.xml -fn
```

Para obter mais informações, consulte [Sintaxe de buildspec](#).

Marcar um grupo de relatórios em AWS CodeBuild

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. Cada AWS tag tem duas partes:

- Uma chave de tag (por exemplo CostCenter, Environment, Project ou Secret). Chaves de tag fazem distinção entre maiúsculas e 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. Como chaves de tag, os valores das tags diferenciam maiúsculas de minúsculas.

Juntos, esses são conhecidos como pares de chave-valor. Para limites no número de tags que é possível ter em um repositório e restrições nos valores e chaves de tags, consulte [Tags](#).

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 grupo de CodeBuild relatórios que você atribui a um bucket do Amazon S3. Para obter mais informações sobre o uso de tags, consulte o informe [Práticas recomendadas de marcação](#).

Em CodeBuild, os recursos principais são o grupo de relatórios e o projeto. Você pode usar o CodeBuild console, o AWS CLI CodeBuild APIs, ou AWS SDKs para adicionar, gerenciar e remover tags de um grupo de relatórios. Além de identificar, organizar e monitorar o grupo de relatórios com tags, é possível usar tags em políticas do IAM para ajudar a controlar quem pode visualizar e interagir com o grupo de relatórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Tópicos

- [Adicionar tags a um grupo de relatórios](#)
- [Visualizar tags de um grupo de relatórios](#)
- [Editar tags de um grupo de relatórios](#)
- [Remover tags de um grupo de relatórios](#)

Adicionar tags a um grupo de relatórios

Adicionar tags a um grupo de relatórios 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 grupo de relatórios. Lembre-se de que há limites para o número de tags que é possível ter em um grupo de relatórios. Há restrições com relação aos caracteres que é possível usar nos campos chave e valor. Para obter mais informações, consulte [Tags](#). Depois que tiver tags, será possível criar

políticas do IAM para gerenciar o acesso ao grupo de relatórios com base nessas tags. Você pode usar o CodeBuild console ou o AWS CLI para adicionar tags a um grupo de relatórios.

Important

Adicionar tags a um grupo de relatórios pode afetar o acesso a ele. Antes de adicionar uma tag a um grupo de relatórios, revise as políticas do IAM que podem usar tags para controlar o acesso a recursos, como grupos de relatórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Para obter mais informações sobre como adicionar tags a um grupo de relatórios ao criá-lo, consulte [Criar um grupo de relatórios \(console\)](#).

Tópicos

- [Adicionar uma tag a um grupo de relatórios \(console\)](#)
- [Adicionar uma tag a um grupo de relatórios \(AWS CLI\)](#)

Adicionar uma tag a um grupo de relatórios (console)

Você pode usar o CodeBuild console para adicionar uma ou mais tags a um grupo de CodeBuild relatórios.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Grupos de relatórios, escolha o nome do grupo de relatórios ao qual deseja adicionar tags.
3. No painel de navegação, selecione Configurações.
4. Se nenhuma tag tiver sido adicionada ao grupo de relatórios, selecione Adicionar tag. Também é possível escolher Editar e 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 grupo de relatórios (AWS CLI)

Para adicionar uma tag a um grupo de relatórios ao criá-lo, consulte [Criar um grupo de relatórios \(CLI\)](#). Em `CreateReportGroup.json`, adicione as tags.

Para adicionar tags a um grupo de relatórios existente, consulte [Atualizar um grupo de relatórios \(CLI\)](#) e adicione as tags no `UpdateReportGroupInput.json`.

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](#).

Visualizar tags de um grupo de relatórios

As tags podem ajudar você a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Para obter mais informações sobre o uso de tags, consulte o informe [Práticas recomendadas de marcação](#). Para obter exemplos de políticas de acesso baseadas em tags, consulte [Deny or allow actions on report groups based on resource tags](#).

Visualizar tags de um grupo de relatórios (console)

Você pode usar o CodeBuild console para visualizar as tags associadas a um grupo de CodeBuild relatórios.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Grupos de relatórios, escolha o nome do grupo de relatórios no qual deseja visualizar tags.
3. No painel de navegação, selecione Configurações.

Visualizar tags de um grupo de relatórios (AWS CLI)

Siga estas etapas para usar o AWS CLI para visualizar as AWS tags de um grupo de relatórios. Se nenhuma tag tiver sido adicionada, a lista de tags retornará vazia.

1. Use o console ou o AWS CLI para localizar o ARN do seu grupo de relatórios. Tome note dele.

AWS CLI

Execute o seguinte comando:

```
aws list-report-groups
```

Esse comando retorna informações formatadas em JSON semelhantes às seguintes:

```
{
  "reportGroups": [
    "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
  ]
}
```

O ARN de um grupo de relatórios termina com o nome do grupo, que pode ser usado para identificar o ARN do grupo de relatórios.

Console

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
 2. Em Grupos de relatórios, escolha o nome do grupo de relatórios com as tags que deseja visualizar.
 3. Em Configuração localize o ARN do grupo de relatórios.
2. Execute o seguinte comando: Use o ARN que você anotou para o parâmetro `--report-group-arns`.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

Se tiver êxito, esse comando retornará informações formatadas em JSON com uma seção `tags` semelhante à seguinte:

```
{
  ...
  "tags": {
    "Status": "Secret",
    "Project": "TestBuild"
  }
  ...
}
```

Editar tags de um grupo de relatórios

É possível alterar o valor de uma tag associada a um grupo de relatórios. Também é possível alterar o nome da chave, o que é equivalente a excluir a tag atual e adicionar outra com o novo nome e o mesmo valor da outra chave. Lembre-se de que há restrições em relação aos caracteres que podem ser usados nos campos de chave e valor. Para obter mais informações, consulte [Tags](#).

Important

A edição de tags de um grupo de relatórios pode afetar o acesso a ele. Antes de editar o nome (chave) ou o valor de uma tag de um grupo de relatórios, revise as políticas do IAM que podem usar essa chave ou esse valor para uma tag a fim de controlar o acesso a recursos, como grupos de relatórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Deny or allow actions on report groups based on resource tags](#).

Editar uma tag de um grupo de relatórios (console)

Você pode usar o CodeBuild console para editar as tags associadas a um grupo de CodeBuild relatórios.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Grupos de relatórios, escolha o nome do grupo de relatórios do qual deseja editar tags.
3. No painel de navegação, selecione Configurações.
4. Selecione Editar.
5. 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 grupo de relatórios (AWS CLI)

Para adicionar, alterar ou excluir tags de um grupo de relatórios, consulte [Atualizar um grupo de relatórios \(CLI\)](#). Atualize as tags em `UpdateReportGroupInput.json`.

Remover tags de um grupo de relatórios

É possível excluir uma ou mais tags associadas a um grupo de relatórios. A remoção de uma tag não exclui a tag de outros AWS recursos associados a essa tag.

Important

A remoção de tags de um grupo de relatórios pode afetar o acesso a ele. Antes de remover uma tag de um grupo de relatórios, revise as políticas do IAM que podem usar a chave ou o valor para uma tag a fim de controlar o acesso a recursos, como grupos de relatórios. Para obter exemplos de políticas de acesso baseadas em tags, consulte [Usar tags para controlar o acesso aos recursos do AWS CodeBuild](#).

Remover uma tag de um grupo de relatórios (console)

Você pode usar o CodeBuild console para remover a associação entre uma tag e um grupo de CodeBuild relatórios.

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. Em Grupos de relatórios, escolha o nome do grupo de relatórios do qual deseja remover tags.
3. No painel de navegação, selecione Configurações.
4. Selecione Editar.
5. Encontre a tag que deseja excluir e selecione Remover tag.
6. Quando terminar de excluir tags, selecione Enviar.

Remover uma tag de um grupo de relatórios (AWS CLI)

Siga estas etapas para usar o AWS CLI para remover uma tag de um grupo de CodeBuild relatórios. Remover uma tag não a exclui, apenas remove a associação entre a tag e o grupo de relatórios.

Note

Se você excluir um grupo de CodeBuild relatórios, todas as associações de tags serão removidas do grupo de relatórios excluído. Não é necessário remover as tags antes de excluir um grupo de relatórios.

Para excluir uma ou mais tags de um grupo de relatórios, consulte [Editar tags de um grupo de relatórios \(AWS CLI\)](#). Atualize a seção tags nos dados formatados em JSON com uma lista atualizada de tags que não contenha as que você deseja excluir. Se quiser excluir todas as tags, atualize a seção tags para:

```
"tags: []"
```

Atualizar um grupo de relatórios

Ao atualizar um grupo de relatórios, é possível especificar informações sobre se deseja exportar os dados do resultado de teste bruto para arquivos em um bucket do Amazon S3. Se você optar por exportar para um bucket do S3, é possível especificar o seguinte para o grupo de relatórios:

- Se os arquivos de resultados de teste bruto são compactados em um arquivo ZIP.
- Se os arquivos do resultado de teste bruto estão criptografados. Você pode especificar a criptografia com uma das seguintes opções:
 - E Chave gerenciada pela AWS para o Amazon S3.
 - Uma chave gerenciada pelo cliente que você cria e configura.

Para obter mais informações, consulte [Criptografia de dados](#).

Se você usar o AWS CLI para atualizar um grupo de relatórios, também poderá atualizar ou adicionar tags. Para obter mais informações, consulte [Marcar um grupo de relatórios em AWS CodeBuild](#).

Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

Tópicos

- [Atualizar um grupo de relatórios \(console\)](#)
- [Atualizar um grupo de relatórios \(CLI\)](#)

Atualizar um grupo de relatórios (console)

Use o procedimento a seguir para atualizar o grupo de relatórios usando a AWS Management Console.

Para atualizar um grupo de relatórios

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Grupos de relatórios.
3. Escolha o grupo de relatórios que deseja atualizar.
4. Selecione Editar.
5. Selecione ou desmarque Fazer backup para Amazon S3. Caso tenha selecionado essa opção, especifique suas configurações de exportação:
 - a. Em Nome do bucket do S3, insira o nome do bucket do S3.
 - b. Em Prefixo do caminho, insira o caminho no bucket do S3 onde deseja fazer upload dos resultados do teste.
 - c. Selecione Compactar dados do resultado do teste em um arquivo zip para compactar os arquivos de dados do resultado de teste bruto.
 - d. Expanda Configuração adicional para exibir opções de criptografia. Escolha uma das seguintes opções:
 - Chave AWS gerenciada padrão para usar a Chave gerenciada pela AWS para o Amazon S3. Para obter mais informações, consulte [Gerenciado pelo cliente CMKs](#) no Guia AWS Key Management Service do usuário. Esta é a opção de criptografia padrão.
 - Escolher uma chave personalizada para usar uma chave gerenciada pelo cliente criada e configurada por você. Para Chave de criptografia do AWS KMS, insira o ARN da chave de criptografia. O formato é `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-id>`. Para obter mais informações, consulte [Criar chaves do KMS](#) no Guia do usuário do AWS Key Management Service.
 - Desativar criptografia de artefatos para desabilitar a criptografia. Você pode escolher isso se quiser compartilhar seus resultados de teste ou publicá-los em um site estático. (Um site dinâmico pode executar códigos para descriptografar os resultados de teste.)

Atualizar um grupo de relatórios (CLI)

Use o procedimento a seguir para atualizar o grupo de relatórios usando a AWS CLI.

Para atualizar um grupo de relatórios

1. Crie um arquivo chamado `UpdateReportGroupInput.json`.
2. Copie o seguinte em `UpdateReportGroupInput.json`:

```
{
  "arn": "",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "bucket-name",
      "path": "path",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "your-key"
    }
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```

3. Insira o ARN do grupo de relatórios na linha do `arn` (por exemplo, `"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1"`).
4. Atualize `UpdateReportGroupInput.json` com as atualizações que deseja aplicar ao grupo de relatórios.
 - Se você deseja atualizar o grupo de relatórios para exportar arquivos de resultados de teste brutos para um bucket do S3, atualize a seção `exportConfig`. Substitua `bucket-name` pelo nome do bucket do S3 e `path` pelo caminho no bucket do S3 para o qual você deseja exportar os arquivos. Se desejar compactar os arquivos exportados, para `packaging`, especifique `ZIP`. Caso contrário, especifique `NONE`. Use `encryptionDisabled` para

especificar se deseja criptografar os arquivos exportados. Se você criptografar os arquivos exportados, insira a chave gerenciada pelo cliente.

- Se você deseja atualizar o grupo de relatórios para que ele não exporte os arquivos de resultado de teste brutos para um bucket do S3, atualize a seção `exportConfig` com o seguinte JSON:

```
{
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

- Se quiser atualizar as tags do grupo de relatórios, atualize a seção `tags`. É possível alterar, adicionar ou remover tags. Se quiser remover todas as tags, atualize-o com o seguinte JSON:

```
"tags": []
```

5. Execute o seguinte comando:

```
aws codebuild update-report-group \
--cli-input-json file:///UpdateReportGroupInput.json
```

Framework de teste

Os tópicos desta seção demonstram como configurar relatórios de teste AWS CodeBuild para várias estruturas de teste.

Tópicos

- [Configurar relatórios de teste com Jasmine](#)
- [Configurar relatórios de teste com Jest](#)
- [Configurar relatórios de teste com pytest](#)
- [Configure relatórios de teste com RSpec](#)

Configurar relatórios de teste com Jasmine

O procedimento a seguir demonstra como configurar relatórios de teste AWS CodeBuild com a estrutura de teste [JasmineBDD](#).

O procedimento requer os seguintes pré-requisitos:

- Você tem um CodeBuild projeto existente.
- O projeto é Node.js configurado para usar a estrutura de trabalho de teste Jasmine.

Adicione o pacote [jasmine-reporters](#) à seção `devDependencies` do arquivo `package.json` do projeto. Este pacote tem uma coleção de classes de JavaScript repórter que podem ser usadas com Jasmine.

```
npm install --save-dev jasmine-reporters
```

Se ainda não estiver presente, adicione o script `test` ao arquivo `package.json` do projeto. O script `test` garante que o Jasmine seja chamado quando `npm test` for executado.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild suporta os seguintes repórteres do teste Jasmine:

JUnitXmlReporter

Usado para gerar relatórios no formato `JUnitXml`.

NUnitXmlReporter

Usado para gerar relatórios no formato `NUnitXml`.

Um projeto Node.js com Jasmine terá, por padrão, um subdiretório `spec`, que contém a configuração da Jasmine e scripts de teste.

Para configurar o Jasmine para gerar relatórios no formato `JUnitXML`, instancie o relatório `JUnitXmlReporter` adicionando o código a seguir aos testes.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
```

```
    savePath: <test report directory>,  
    filePrefix: <report filename>,  
    consolidateAll: true  
  });  
  
jasmine.getEnv().addReporter(junitReporter);
```

Para configurar o Jasmine para gerar relatórios no formato NUnitXML, instancie o relatório `NUnitXmlReporter` adicionando o código a seguir aos testes.

```
var reporters = require('jasmine-reporters');  
  
var nunitReporter = new reporters.NUnitXmlReporter({  
  savePath: <test report directory>,  
  filePrefix: <report filename>,  
  consolidateAll: true  
});  
  
jasmine.getEnv().addReporter(nunitReporter)
```

Os relatórios de teste são exportados para o arquivo especificado por `<test report directory>/<report filename>`.

No arquivo `buildspec.yml`, adicione/atualize as seções a seguir.

```
version: 0.2  
  
phases:  
  pre_build:  
    commands:  
      - npm install  
  build:  
    commands:  
      - npm build  
      - npm test  
  
reports:  
  jasmine_reports:  
    files:  
      - <report filename>  
    file-format: JUNITXML  
    base-directory: <test report directory>
```

Se você estiver usando o formato de relatório NunitXml, altere o valor `file-format` para o seguinte.

```
file-format: NUNITXML
```

Configurar relatórios de teste com Jest

O procedimento a seguir demonstra como configurar os relatórios de teste AWS CodeBuild com a estrutura de [testes do Jest](#).

O procedimento requer os seguintes pré-requisitos:

- Você tem um CodeBuild projeto existente.
- O projeto é um projeto Node.js configurado para usar a estrutura de trabalho de teste Jest.

Adicione o [jest-junit](#) pacote à `devDependencies` seção do `package.json` arquivo do seu projeto. CodeBuild usa esse pacote para gerar relatórios no JunitXml formato.

```
npm install --save-dev jest-junit
```

Se ainda não estiver presente, adicione o script `test` ao arquivo `package.json` do projeto. O script `test` garante que o Jest seja chamado quando `npm test` for executado.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Configure a Jest para usar o relatório JunitXml adicionando o seguinte ao arquivo de configuração da Jest. Se o projeto não tiver um arquivo de configuração da Jest, crie um arquivo chamado `jest.config.js` na raiz do projeto e adicione o seguinte. Os relatórios de teste são exportados para o arquivo especificado por `<test report directory>/<report filename>`.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
```

```
    outputDirectory: <test report directory>,  
    outputName: <report filename>,  
  } ]  
]  
};
```

No arquivo `buildspec.yml`, adicione/atualize as seções a seguir.

```
version: 0.2  
  
phases:  
  pre_build:  
    commands:  
      - npm install  
  build:  
    commands:  
      - npm build  
      - npm test  
  
reports:  
  jest_reports:  
    files:  
      - <report filename>  
    file-format: JUNITXML  
    base-directory: <test report directory>
```

Configurar relatórios de teste com pytest

O procedimento a seguir demonstra como configurar relatórios de teste AWS CodeBuild com a estrutura de [teste pytest](#).

O procedimento requer os seguintes pré-requisitos:

- Você tem um CodeBuild projeto existente.
- O projeto é um projeto Python configurado para usar a estrutura de trabalho de teste pytest.

Adicione a seguinte entrada à fase `build` ou `post_build` do arquivo `buildspec.yml`. Esse código descobre automaticamente os testes no diretório atual e exporta os relatórios de teste para o arquivo especificado por `<test report directory>/<report filename>`. O relatório usa o formato `JunitXml`.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

No arquivo `buildspec.yml`, adicione/atualize as seções a seguir.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Configure relatórios de teste com RSpec

O procedimento a seguir demonstra como configurar relatórios de teste AWS CodeBuild com a [estrutura de RSpec teste](#).

O procedimento requer os seguintes pré-requisitos:

- Você tem um CodeBuild projeto existente.
- Seu projeto é um projeto Ruby configurado para usar a estrutura de RSpec teste.

Adicione/atualize o seguinte no arquivo `buildspec.yml`. Esse código executa os testes no `<test source directory>` diretório e exporta os relatórios de teste para o arquivo especificado por `<test report directory>/<report filename>`. O relatório usa o formato JunitXml.

```
version: 0.2
```



```
phases:
  install:
    runtime-versions:
      ruby: 2.6
  pre_build:
    commands:
      - gem install rspec
      - gem install rspec_junit_formatter
  build:
    commands:
      - rspec <test source directory>/ * --format RspecJunitFormatter --out <test report
        <test report directory>/<report filename>
  reports:
    rspec_reports:
      files:
        - <report filename>
      base-directory: <test report directory>
      file-format: JUNITXML
```

Visualizar relatórios de teste

Você pode visualizar detalhes sobre um relatório de teste, como informações sobre seus casos de teste, números de aprovação e reprovação e quanto tempo levou para que ele fosse executado.

Você pode visualizar relatórios de teste agrupados por execução de compilação, grupo de relatórios ou sua AWS conta. Escolha um relatório de teste no console para exibir os detalhes e resultados dos seus casos de teste.

É possível visualizar os relatórios de teste de exibição que não estão expirados. Os relatórios de teste expiram 30 dias após sua criação. Não é possível visualizar um relatório expirado no CodeBuild.

Tópicos

- [Visualizar relatórios de teste para uma compilação](#)
- [Visualizar relatórios de teste para um grupo de relatórios](#)
- [Visualizar relatórios de teste em sua conta da AWS](#)

Visualizar relatórios de teste para uma compilação

Como visualizar relatórios de teste para uma compilação

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Localize a compilação que deseja visualizar. Se você souber o projeto que executou a compilação que criou o relatório de teste:
 1. No painel de navegação, escolha Build projects (Projetos de compilação) e selecione o projeto com a compilação que executou o relatório de teste que você deseja visualizar.
 2. Escolha Build history (Histórico de compilação) e a compilação executada que criou os relatórios que você deseja visualizar.

Você também pode localizar a compilação no histórico de compilação da sua conta da AWS :

1. No painel de navegação, escolha Build history (Histórico de compilação) e a compilação que criou os relatórios de teste que você deseja visualizar.
3. Na página de compilação, escolha Reports (Relatórios) e escolha um relatório de teste para exibir detalhes.

Visualizar relatórios de teste para um grupo de relatórios

Como visualizar relatórios de teste em um grupo de relatórios

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, escolha Grupos de relatórios.
3. Escolha o grupo de relatórios que contém os relatórios de teste que você deseja visualizar.
4. Escolha um relatório de teste para exibir detalhes.

Visualizar relatórios de teste em sua conta da AWS

Para ver os relatórios de teste em sua AWS conta

1. Abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Report history (Histórico de relatórios).

3. Escolha um relatório de teste para exibir detalhes.

Permissões de relatório de teste

Este tópico descreve informações importantes sobre permissões relativas aos relatórios de testes.

Tópicos

- [Perfil do IAM para relatórios de testes](#)
- [Permissões para operações de relatórios de testes](#)
- [Exemplos de permissões de relatórios de testes](#)

Perfil do IAM para relatórios de testes

Para executar um relatório de teste e atualizar um projeto para incluir relatórios de teste, sua função do IAM requer as seguintes permissões. Essas permissões estão incluídas nas políticas AWS gerenciadas predefinidas. Caso pretenda adicionar relatórios de teste a um projeto de compilação existente, é necessário adicionar essas permissões por conta própria.

- `CreateReportGroup`
- `CreateReport`
- `UpdateReport`
- `BatchPutTestCases`

Para gerar um relatório de cobertura de código, o perfil do IAM também deve incluir a permissão `BatchPutCodeCoverages`.

Note

`BatchPutTestCases`, `CreateReport`, `UpdateReport` e `BatchPutCodeCoverages` não são permissões públicas. Você não pode chamar um AWS CLI comando ou método SDK correspondente para essas permissões.

Para verificar se você tem essas permissões, é possível anexar a seguinte política ao perfil do IAM:

```
{
```

```
"Effect": "Allow",
"Resource": [
    "*"
],
"Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
]
}
```

Recomendamos que você restrinja essa política apenas aos grupos de relatórios que você precisa usar. O seguinte restringe as permissões somente aos grupos de relatórios com os dois ARNs na política:

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

As opções a seguir restringem permissões apenas para grupos de relatórios criados pela execução de compilações de um projeto chamado `my-project`:

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
  ],
}
```

```
"Action": [  
  "codebuild:CreateReportGroup",  
  "codebuild:CreateReport",  
  "codebuild:UpdateReport",  
  "codebuild:BatchPutTestCases",  
  "codebuild:BatchPutCodeCoverages"  
]  
}
```

Note

A função CodeBuild de serviço especificada no projeto é usada para obter permissões de upload para o bucket do S3.

Permissões para operações de relatórios de testes

Você pode especificar permissões para as seguintes operações da CodeBuild API de relatórios de teste:

- BatchGetReportGroups
- BatchGetReports
- CreateReportGroup
- DeleteReportGroup
- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

Para obter mais informações, consulte [AWS CodeBuild referência de permissões](#).

Exemplos de permissões de relatórios de testes

Para obter informações sobre políticas de exemplo relativas aos relatórios de testes, consulte o seguinte:

- [Permitir que um usuário altere um grupo de relatórios](#)
- [Permitir que um usuário crie um grupo de relatórios](#)
- [Permitir que um usuário exclua um relatório](#)
- [Permitir que um usuário exclua um grupo de relatórios](#)
- [Permitir que um usuário obtenha informações sobre grupos de relatórios](#)
- [Permitir que um usuário obtenha informações sobre relatórios](#)
- [Permitir que um usuário obtenha uma lista de grupos de relatórios](#)
- [Permitir que um usuário obtenha uma lista de relatórios](#)
- [Permitir que um usuário obtenha uma lista de relatórios para um grupo de relatórios](#)
- [Permitir que um usuário obtenha uma lista de casos de teste para um relatório](#)

Status do relatório de teste

O status de um relatório de teste pode ser um dos seguintes:

- **GENERATING**: a execução dos casos de teste ainda está em andamento.
- **DELETING**: o relatório de teste está sendo excluído. Quando um relatório de teste é excluído, seus casos de teste também são excluídos. Os arquivos de dados de resultado de teste bruto exportados para um bucket do S3 não são excluídos.
- **INCOMPLETE**: o relatório de teste não foi concluído. Esse status pode ser retornado por um dos seguintes motivos:
 - Um problema com a configuração do grupo de relatórios que especifica os casos de teste deste relatório. Por exemplo, o caminho para os casos de teste no grupo de relatórios do arquivo buildspec pode estar incorreto.
 - O usuário do IAM que executou a compilação não tem permissões para executar testes. Para obter mais informações, consulte [Permissões de relatório de teste](#).
 - A compilação não foi concluída devido a um erro que não está relacionado aos testes.
- **SUCCEEDED**: todos os casos de teste foram bem-sucedidos.
- **FAILED**: alguns casos de teste não foram bem-sucedidos.

Cada caso de teste retorna um status. O status de um caso de teste pode ser um dos seguintes:

- **SUCCEEDED**: o caso de teste foi aprovado.

- **FAILED:** ocorreu uma falha no caso de teste.
- **ERROR:** ocorreu um erro inesperado no caso de teste.
- **SKIPPED:** o caso de teste não foi executado.
- **UNKNOWN:** o caso de teste retornou um status diferente de **SUCCEEDED**, **FAILED**, **ERROR** ou **SKIPPED**.

Um relatório de teste pode ter, no máximo, 500 resultados de caso de teste. Se mais de 500 casos de teste forem executados, CodeBuild priorizará os testes com o status **FAILED** e truncará os resultados do caso de teste.

Use AWS CodeBuild com a Amazon Virtual Private Cloud

Normalmente, AWS CodeBuild não é possível acessar recursos em uma VPC. Para habilitar o acesso, você deve fornecer informações adicionais de configuração específicas da VPC na configuração do seu CodeBuild projeto. Isso inclui o ID da VPC, a sub-rede da VPC IDs e o grupo de segurança da VPC. IDs As compilações habilitadas para VPC podem acessar recursos dentro da VPC. Para obter mais informações sobre como configurar uma VPC na Amazon VPC, consulte o [Guia do usuário da Amazon VPC](#).

Tópicos

- [Casos de uso](#)
- [Melhores práticas para VPCs](#)
- [Limitações do VPCs](#)
- [Permita o acesso da Amazon VPC em seus projetos CodeBuild](#)
- [Solucionar problemas de configuração da VPC](#)
- [Usar VPC endpoints](#)
- [Use AWS CodeBuild com um servidor proxy gerenciado](#)
- [Use AWS CodeBuild com um servidor proxy](#)
- [AWS CloudFormation Modelo de VPC](#)

Casos de uso

A conectividade VPC a partir de AWS CodeBuild compilações possibilita:

- Executar testes de integração na compilação em relação a dados em um banco de dados do Amazon RDS isolado em uma sub-rede privada.
- Consulte dados em um ElastiCache cluster da Amazon diretamente dos testes.
- Interaja com serviços web internos hospedados na Amazon EC2, Amazon ECS ou serviços que usam o Elastic Load Balancing interno.
- Recuperar dependências de repositórios de artefatos auto-hospedados, internos, como PyPI para Python, Maven para Java e npm para Node.js.
- Acessar objetos em um bucket do S3 configurado para permitir acesso apenas por meio de um endpoint da VPC da Amazon.

- Consultar serviços da web externos que exijam endereços IP fixos por meio do endereço IP elástico do gateway NAT ou da instância NAT associados à sub-rede.

As compilações podem acessar qualquer recurso hospedado na VPC.

Melhores práticas para VPCs

Use essa lista de verificação ao configurar uma VPC com a qual trabalhar. CodeBuild

- Configure a VPC com sub-redes públicas e privadas e um gateway NAT. O gateway NAT deve estar localizado em uma sub-rede pública. Para obter mais informações, consulte [VPC com sub-redes privadas e públicas \(NAT\)](#) no Guia do usuário da Amazon VPC.

Important

Você precisa de um gateway NAT ou instância NAT para usar CodeBuild com sua VPC para que ela CodeBuild possa alcançar endpoints públicos (por exemplo, para executar comandos da CLI ao executar compilações). Você não pode usar o gateway de internet em vez de um gateway NAT ou uma instância NAT porque CodeBuild não oferece suporte à atribuição de endereços IP elásticos às interfaces de rede que ele cria, e a atribuição automática de um endereço IP público não é suportada pela Amazon EC2 para nenhuma interface de rede criada fora da execução de instâncias da Amazon. EC2

- Inclua várias zonas de disponibilidade com a VPC.
- Certifique-se de que seus grupos de segurança não tenham tráfego de entrada (entrada) permitido em suas compilações. CodeBuild não tem requisitos específicos para tráfego de saída, mas você deve permitir o acesso a todos os recursos da Internet necessários para sua construção, como GitHub o Amazon S3.

Para obter mais informações, consulte [Regras de grupos de segurança](#) no Guia do usuário da Amazon VPC.

- Configure sub-redes à parte para as compilações.
- Ao configurar seus CodeBuild projetos para acessar sua VPC, escolha somente sub-redes privadas.

Para obter mais informações sobre como configurar uma VPC na Amazon VPC, consulte o [Guia do usuário da Amazon VPC](#).

Para obter mais informações sobre AWS CloudFormation como configurar uma VPC para usar o recurso CodeBuild VPC, consulte o [AWS CloudFormation Modelo de VPC](#)

Limitações do VPCs

- A conectividade VPC de não CodeBuild é compatível com compartilhamento. VPCs

Permita o acesso da Amazon VPC em seus projetos CodeBuild

Inclua essas configurações na definição do VPC:

- Para VPC ID, escolha a VPC ID que usa. CodeBuild
- Para sub-redes, escolha uma sub-rede privada com tradução NAT que inclua ou tenha rotas para os recursos usados pelo. CodeBuild
- Em Grupos de Segurança, escolha os grupos de segurança CodeBuild usados para permitir o acesso aos recursos no VPCs.

Para usar o console para criar um projeto de compilação, consulte [Criar um projeto de compilação \(console\)](#). Ao criar ou alterar seu CodeBuild projeto, na VPC, escolha seu ID de VPC, sub-redes e grupos de segurança.

Para usar o AWS CLI para criar um projeto de construção, consulte [Criar um projeto de compilação \(AWS CLI\)](#). Se você estiver usando o AWS CLI with CodeBuild, a função de serviço usada por CodeBuild para interagir com os serviços em nome do usuário do IAM deve ter uma política anexada. Para mais informações, consulte [Permitir CodeBuild acesso aos AWS serviços necessários para criar uma interface de rede VPC](#).

O *vpcConfig* objeto deve incluir seu *vpcIdsecurityGroupIds*, *subnets* e.

- *vpcId*: Obrigatório. O ID da VPC que CodeBuild usa. Execute este comando para obter uma lista de todas as Amazon VPC IDs em sua região:

```
aws ec2 describe-vpcs
```

- *subnets*: Obrigatório. A sub-rede IDs que inclui recursos usados pelo CodeBuild. Execute este comando e obtenha estes IDs:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Substitua `us-east-1` pela sua região.

- ***securityGroupIds***: Obrigatório. O grupo de segurança IDs usado por CodeBuild para permitir o acesso aos recursos no VPCs. Execute este comando para obter estes IDs:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Substitua `us-east-1` pela sua região.

Solucionar problemas de configuração da VPC

Use as informações exibidas na mensagem de erro para ajudar a identificar, diagnosticar e resolver problemas.

Estão são algumas diretrizes para auxiliar na solução de problemas de um erro da VPC do CodeBuild comum: `Build does not have internet connectivity. Please check subnet network configuration.`

1. [Certifique-se de que o gateway da Internet esteja anexado à VPC.](#)
2. [Certifique-se de que a tabela de rotas para a sub-rede pública aponte para o gateway da Internet.](#)
3. [Certifique-se de que sua rede ACLs permita que o tráfego flua.](#)
4. [Certifique-se de que os grupos de segurança permitam o fluxo do tráfego.](#)
5. [Solucione problemas do gateway NAT.](#)
6. [Certifique-se de que a tabela de rotas para sub-redes privadas aponte para o gateway NAT.](#)

7. Certifique-se de que a função de serviço usada por CodeBuild para interagir com os serviços em nome do usuário do IAM tenha as permissões [nesta política](#). Para obter mais informações, consulte [CodeBuild Permitir interagir com outros AWS serviços](#).

Se CodeBuild faltarem permissões, você pode receber um erro que diz, `Unexpected EC2 error: UnauthorizedOperation`. Esse erro pode ocorrer se você CodeBuild não tiver as EC2 permissões da Amazon necessárias para trabalhar com uma VPC.

Usar VPC endpoints

Você pode melhorar a segurança de suas compilações configurando AWS CodeBuild para usar uma interface VPC endpoint. Os endpoints de interface são alimentados por PrivateLink uma tecnologia que você pode usar para acessar a Amazon de forma privada EC2 e CodeBuild usar endereços IP privados. PrivateLink restringe todo o tráfego de rede entre suas instâncias CodeBuild gerenciadas e EC2 a Amazon na rede Amazon. (As instâncias gerenciadas não têm acesso à Internet.) Além disso, você não precisa de um gateway da Internet, de um dispositivo NAT ou de um gateway privado virtual. Não é necessário configurar o PrivateLink, mas é recomendável. [Para obter mais informações sobre PrivateLink endpoints de VPC, consulte O que é? AWS PrivateLink](#).

Antes de criar VPC endpoints

Antes de configurar os VPC endpoints para AWS CodeBuild, esteja ciente das seguintes restrições e limitações.

Note

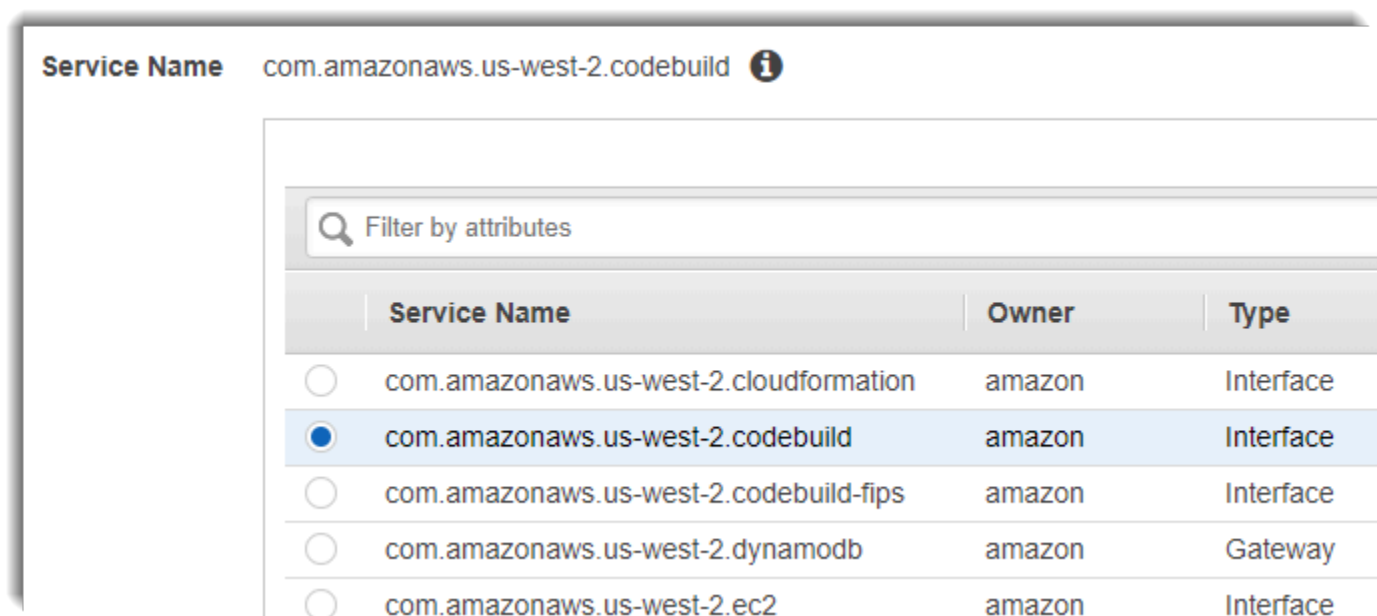
Use um [gateway NAT](#) se quiser usar CodeBuild com AWS serviços que não suportam conexões Amazon PrivateLink VPC.

- Os endpoints da VPC são compatíveis somente com DNS fornecido pela Amazon por meio do Amazon Route 53. Se quiser usar seu próprio DNS, poderá usar o encaminhamento de DNS condicional. Para obter mais informações, consulte [Conjuntos de opções DHCP](#) no Guia do usuário do Amazon VPC.
- Atualmente, os VPC endpoints não oferecem suporte a solicitações entre regiões. Certifique-se de criar seu endpoint na mesma AWS região de qualquer bucket do S3 que armazene sua entrada e saída de compilação. Você pode usar o console do Amazon S3 ou o [get-bucket-location](#) comando

para encontrar a localização do seu bucket. Use um endpoint do Amazon S3 específico da região para acessar o bucket (por exemplo, `<bucket-name>.s3-us-west-2.amazonaws.com`). Para obter mais informações sobre endpoints específicos da região para o Amazon S3, consulte [Amazon Simple Storage Service](#) na Referência geral da Amazon Web Services. Se você usar o AWS CLI para fazer solicitações ao Amazon S3, defina sua região padrão para a mesma região em que seu bucket foi criado ou use o `--region` parâmetro em suas solicitações.

Crie endpoints VPC para CodeBuild

Siga as instruções em [Criar um endpoint de interface](#) para criar o endpoint `com.amazonaws.region.codebuild`. Esse é um VPC endpoint para AWS CodeBuild



`region` representa o identificador de uma AWS região suportada por CodeBuild, como `us-east-2` a região Leste dos EUA (Ohio). Para obter uma lista das AWS regiões suportadas, consulte [CodeBuild](#) na Referência AWS geral. O endpoint é pré-preenchido com a região que você especificou quando fez login. AWS Se você alterar a região, o VPC endpoint será atualizado adequadamente.

Crie uma política de VPC endpoint para CodeBuild

Você pode criar uma política para endpoints da Amazon VPC, AWS CodeBuild 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.

O exemplo de política a seguir especifica que todos os principais só podem iniciar e visualizar compilações para o projeto `project-name`.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Para obter mais informações, consulte [Controlar o acesso a serviços com endpoints da VPC](#) no Guia do Usuário do Amazon VPC.

Use AWS CodeBuild com um servidor proxy gerenciado

Para executar frotas de capacidade AWS CodeBuild reservada em um servidor proxy gerenciado, você deve configurar o servidor proxy para permitir ou negar tráfego de e para sites externos usando regras de proxy. Observe que a execução de frotas de capacidade reservada em um servidor proxy gerenciado não é compatível com VPC, Windows ou macOS.

Important

Há custos adicionais com base na duração em que uma configuração de proxy fica presente na frota. Para obter mais informações, consulte <https://aws.amazon.com/codebuild/preços/>.

Tópicos

- [Definir uma configuração de proxy gerenciado para frotas de capacidade reservada](#)
- [Administre uma frota de capacidade CodeBuild reservada](#)

Definir uma configuração de proxy gerenciado para frotas de capacidade reservada

Para configurar um servidor proxy gerenciado para sua frota de capacidade reservada, você deve habilitar este recurso ao criar a frota no console ou ao usar a AWS CLI. Há várias propriedades que precisam ser definidas:

Definir configurações de proxy - opcional

Configurações de proxy que aplicam controle de acesso à rede às instâncias de capacidade reservada.

Comportamento padrão

Define o comportamento do tráfego de saída.

Permitir

Permite tráfego de saída para todos os destinos por padrão.

Negar

Nega tráfego de saída para todos os destinos por padrão.

Regras de proxy

Especifica os domínios de destino aos quais restringir o controle de acesso à rede.

Para definir as configurações de proxy no console, consulte [Criar uma frota de capacidade reservada](#) para obter instruções. Para definir configurações de proxy usando o AWS CLI, você pode fazer isso modificando a seguinte sintaxe JSON e salvando seus resultados:

```
"proxyConfiguration": {
  "defaultBehavior": "ALLOW_ALL" | "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN" | "IP",
      "effect": "ALLOW" | "DENY",
      "entities": [
        "destination"
      ]
    }
  ]
}
```

```
]
}
```

O arquivo JSON pode ser semelhante ao seguinte:

```
"proxyConfiguration": {
  "defaultBehavior": "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN",
      "effect": "ALLOW",
      "entities": [
        "github.com"
      ]
    }
  ]
}
```

Administre uma frota de capacidade CodeBuild reservada

Ao executar frotas de capacidade AWS CodeBuild reservada com seu servidor proxy gerenciado, CodeBuild definirá automaticamente suas variáveis HTTP_PROXY e de HTTPS_PROXY ambiente com os endereços de proxy gerenciados. Se o software de dependência tiver sua própria configuração e não aderir às variáveis de ambiente, você poderá consultar esses valores e atualizar a configuração do software nos comandos de compilação para rotear adequadamente o tráfego de compilação por meio do proxy gerenciado. Para obter mais informações, consulte [Crie um projeto de construção em AWS CodeBuild](#) e [Alterar as configurações do projeto de construção em AWS CodeBuild](#).

Use AWS CodeBuild com um servidor proxy

Você pode usar AWS CodeBuild com um servidor proxy para regular o tráfego HTTP e HTTPS de e para a Internet. Para executar CodeBuild com um servidor proxy, você instala um servidor proxy em uma sub-rede pública e CodeBuild em uma sub-rede privada em uma VPC.

Há dois casos de uso principais para execução CodeBuild em um servidor proxy:

- Ele elimina o uso de um gateway NAT ou de uma instância NAT na sua VPC.
- Ele permite que você especifique o URLs que as instâncias no servidor proxy podem acessar e o URLs que o servidor proxy nega acesso.

Você pode usar CodeBuild com dois tipos de servidores proxy. Para ambos, o servidor proxy é executado em uma sub-rede pública e CodeBuild em uma sub-rede privada.

- Proxy explícito: se você usar um servidor proxy explícito, deverá configurar NO_PROXY e as variáveis de HTTPS_PROXY ambiente no CodeBuild nível do projeto. HTTP_PROXY Para obter mais informações, consulte [Alterar as configurações do projeto de construção em AWS CodeBuild](#) e [Crie um projeto de construção em AWS CodeBuild](#).
- Proxy transparente: se você usar um servidor de proxy transparente, nenhuma configuração especial será necessária.

Tópicos

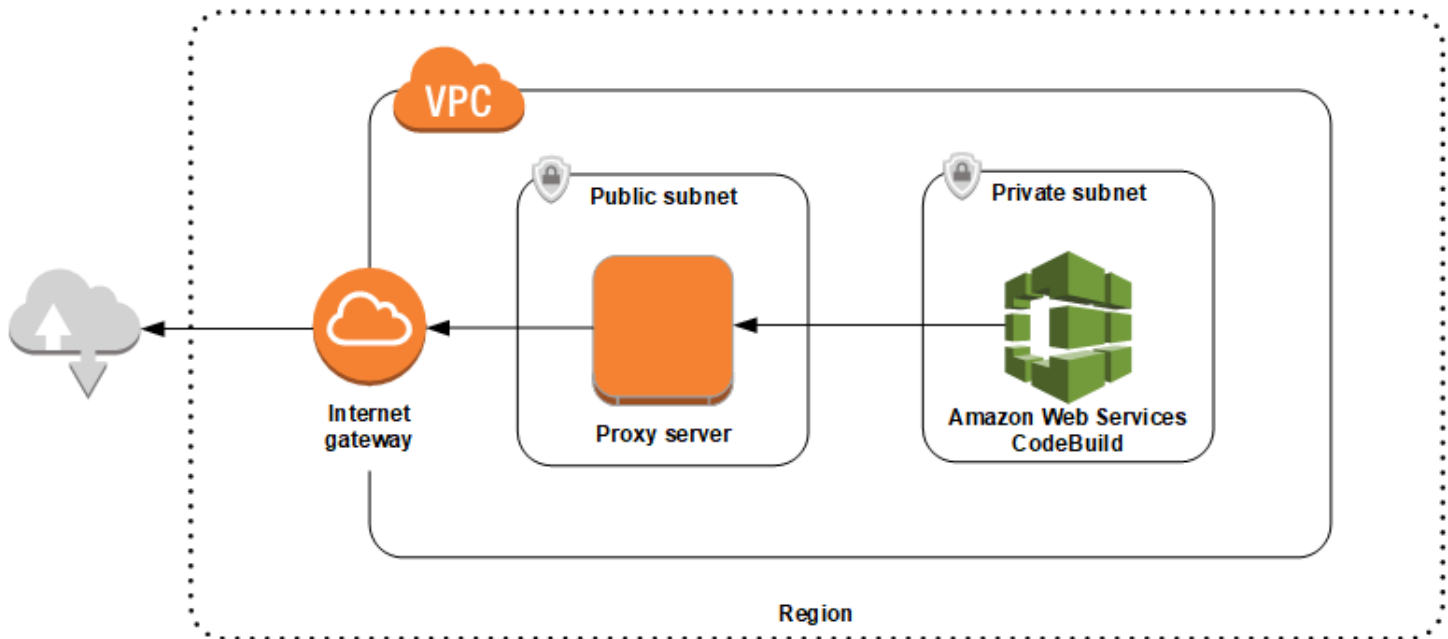
- [Configurar componentes necessários para execução CodeBuild em um servidor proxy](#)
- [Execute CodeBuild em um servidor proxy explícito](#)
- [Execute CodeBuild em um servidor proxy transparente](#)
- [Executar um gerenciador de pacotes e outras ferramentas em um servidor de proxy](#)

Configurar componentes necessários para execução CodeBuild em um servidor proxy

Você precisa que esses componentes sejam executados AWS CodeBuild em um servidor proxy transparente ou explícito:

- Uma VPC.
- Uma sub-rede pública em sua VPC para o servidor de proxy.
- Uma sub-rede privada em sua VPC para o CodeBuild.
- Um gateway da Internet que permite a comunicação entre a VPC e a Internet.

O diagrama a seguir mostra como os componentes interagem.



Configurar uma VPC, sub-redes e um gateway de rede

As etapas a seguir são necessárias para serem executadas AWS CodeBuild em um servidor proxy transparente ou explícito.

1. Crie uma VPC. Para obter informações, consulte [Criar uma VPC](#) no Guia do usuário da Amazon VPC.
2. Crie duas sub-redes em sua VPC. Uma é a sub-rede pública chamada Public Subnet na qual o servidor de proxy é executado. A outra é uma sub-rede privada chamada Private Subnet na qual CodeBuild é executada.

Para mais informações, consulte [Criar uma sub-rede na VPC](#).

3. Crie e vincule um gateway da Internet à VPC. Para mais informações, consulte [Criar e anexar um gateway da Internet](#).
4. Adicione uma regra à tabela de rotas padrão que roteia o tráfego de saída da VPC (0.0.0.0/0) para o gateway da Internet. Para mais informações, consulte [Adicionar e remover rotas de uma tabela de rotas](#).
5. Adicione uma regra ao grupo de segurança padrão da sua VPC que permita o tráfego SSH de entrada (TCP 22) proveniente da sua VPC (0.0.0.0/0).
6. Siga as instruções em [Iniciar uma instância usando o assistente de inicialização de instância](#) no Guia EC2 do usuário da Amazon para iniciar uma instância do Amazon Linux. Quando você executar o assistente, escolha as seguintes opções:

- Em Escolher um tipo de instância, escolha uma Imagem de máquina da Amazon (AMI) do Amazon Linux.
- Em Subnet (Sub-rede), escolha a sub-rede pública criada anteriormente neste tópico. Se você tiver usado o nome sugerido, trata-se da Public Subnet (Sub-rede pública).
- Em Auto-assign Public IP (Atribuir automaticamente o IP público), selecione Enable (Habilitar).
- Na página Configure Security Group (Configurar grupo de segurança), para Assign a security group (Atribuir um grupo de segurança), escolha a opção Select an existing security group (Selecionar um grupo de segurança existente). Em seguida, escolha o grupo de segurança padrão.
- Depois de escolher Launch (Executar), selecione um par de chaves existente ou crie um.

Escolha as configurações padrão para todas as outras opções.

7. Depois que sua EC2 instância estiver em execução, desative as verificações de origem/destino. Para obter informações, consulte [Desabilitar verificações de origem/destino](#) no Guia do usuário da Amazon VPC.
8. Crie uma tabela de rotas na sua VPC. Adicione uma regra à tabela de rotas que roteie para o seu servidor de proxy o tráfego destinado à Internet. Associe essa tabela de rotas à sua sub-rede privada. Isso é necessário para que as solicitações de saída de instâncias em sua sub-rede privada, onde são CodeBuild executadas, sejam sempre roteadas pelo servidor proxy.

Instalar e configurar um servidor de proxy

Há muitos servidores de proxy disponíveis para escolher. Um servidor proxy de código aberto, o Squid, é usado aqui para demonstrar como é AWS CodeBuild executado em um servidor proxy. Você pode aplicar os mesmos conceitos a outros servidores de proxy.

Para instalar o Squid, use um repositório yum executando os seguintes comandos:

```
sudo yum update -y
sudo yum install -y squid
```

Depois que instalar o Squid, edite seu arquivo `squid.conf` usando as instruções mais adiante neste tópico.

Configurar o Squid para tráfego HTTPS

Para HTTPS, o tráfego HTTP é encapsulado em uma conexão Transport Layer Security (TLS – Segurança da camada de transporte). O Squid usa um recurso chamado [SslPeekAndSplice](#) para recuperar a Indicação de Nome do Servidor (SNI) da iniciação TLS que contém o host de Internet solicitado. Isso é necessário para que o Squid não precise descriptografar o tráfego HTTPS. Para habilitar SslPeekAndSplice, o Squid exige um certificado. Crie esse certificado usando OpenSSL:

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

Para HTTP, o Squid não requer configuração. De todas as mensagens de solicitação HTTP/1.1, ele pode recuperar o campo de cabeçalho do host, que especifica o host da Internet que está sendo solicitado.

Execute CodeBuild em um servidor proxy explícito

Para ser executado AWS CodeBuild em um servidor proxy explícito, você deve configurar o servidor proxy para permitir ou negar tráfego de e para sites externos e, em seguida, configurar as HTTP_PROXY variáveis de HTTPS_PROXY ambiente e.

Tópicos

- [Configurar o Squid como um servidor de proxy explícito](#)
- [Criar um projeto do CodeBuild](#)
- [Arquivo squid.conf de exemplo do servidor de proxy explícito](#)

Configurar o Squid como um servidor de proxy explícito

Para configurar o servidor de proxy do Squid como explícito, você deve fazer as seguintes modificações no arquivo `/etc/squid/squid.conf` dele:

- Remova as seguintes regras da lista de controle de acesso (ACL) padrão.

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Adicione a seguinte informação no lugar das regras da ACL padrão que você removeu. A primeira linha permite solicitações da sua VPC. As próximas duas linhas concedem ao servidor proxy acesso ao destino URLs que pode ser usado por AWS CodeBuild. Edite a expressão regular na última linha para especificar buckets do S3 ou um CodeCommit repositório em uma região. AWS Por exemplo:

- Se a origem for o Amazon S3, use o comando `acl download_src dstdom_regex .*s3\.us-west-1\.amazonaws\.com` para conceder acesso aos buckets do S3 na região `us-west-1`.
- Se sua fonte for AWS CodeCommit, use `git-codecommit.<your-region>.amazonaws.com` para adicionar uma AWS região a uma lista de permissões.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from
Amazon S3 or CodeCommit
```

- Substitua `http_access allow localnet` pelo seguinte:

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Se quiser que a compilação faça upload de logs e artefatos, siga um destes procedimentos:
 1. Antes da instrução `http_access deny all`, insira as instruções a seguir. Eles permitem CodeBuild acessar CloudWatch o Amazon S3. O acesso a CloudWatch é necessário para que CodeBuild possa criar CloudWatch registros. O acesso ao Amazon S3 é necessário para fazer upload de artefatos e o armazenamento em cache do Amazon S3.

```
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
```

```
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

- Depois de salvar `squid.conf`, execute o seguinte comando:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Adicione proxy ao arquivo `buildspec`. Para obter mais informações, consulte [Sintaxe de buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
      - command
```

Note

Se você receber um erro de RequestError tempo limite, consulte [RequestError erro de tempo limite ao executar CodeBuild em um servidor proxy](#).

Para obter mais informações, consulte [Arquivo squid.conf de exemplo do servidor de proxy explícito](#) mais adiante neste tópico.

Criar um projeto do CodeBuild

Para executar AWS CodeBuild com seu servidor proxy explícito, defina suas variáveis `HTTP_PROXY` e de `HTTPS_PROXY` ambiente com o endereço IP privado da EC2 instância que você criou para seu servidor proxy e a porta 3128 no nível do projeto. O endereço IP privado é semelhante a `http://your-ec2-private-ip-address:3128`. Para obter mais informações, consulte [Alterar](#)

[as configurações do projeto de construção em AWS CodeBuild](#) e [Crie um projeto de construção em AWS CodeBuild](#).

Use o comando a seguir para visualizar o log de acesso do proxy do Squid:

```
sudo tail -f /var/log/squid/access.log
```

Arquivo **squid.conf** de exemplo do servidor de proxy explícito

Veja a seguir um exemplo de um arquivo squid.conf configurado para um servidor de proxy explícito.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
# add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com    #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80  # http
acl Safe_ports port 21  # ftp
acl Safe_ports port 443  # https
acl Safe_ports port 70  # gopher
acl Safe_ports port 210  # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280  # http-mgmt
acl Safe_ports port 488  # gss-http
acl Safe_ports port 591  # filemaker
acl Safe_ports port 777  # multiling http
acl CONNECT method CONNECT
#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
```

```
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
# Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
# And finally deny all other access to this proxy
http_access deny all
# Squid normally listens to port 3128
http_port 3128
# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```


Execute CodeBuild em um servidor proxy transparente

Para ser executado AWS CodeBuild em um servidor proxy transparente, você deve configurar o servidor proxy com acesso aos sites e domínios com os quais ele interage.

Tópicos

- [Configurar o Squid como um servidor de proxy transparente](#)
- [Criar um projeto do CodeBuild](#)

Configurar o Squid como um servidor de proxy transparente

Para configurar um servidor de proxy como transparente, você deve conceder a ele o acesso aos domínios e sites que deseja que ele acesse. Para executar AWS CodeBuild com um servidor proxy transparente, você deve conceder acesso a `amazonaws.com`. Você também deve conceder acesso a outros CodeBuild usos de sites. Eles variam, dependendo de como você cria seus CodeBuild projetos. Sites de exemplo são aqueles para repositórios como Bitbucket, GitHub, Yum e Maven. Para conceder ao Squid o acesso a domínios e sites específicos, use um comando semelhante ao mencionado a seguir a fim de atualizar o arquivo `squid.conf`. Este comando de exemplo concede acesso a `amazonaws.com`, `github.com` e `bitbucket.com`. Você pode editar esse exemplo para conceder acesso a outros sites.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
```

```
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
EOF
```

As solicitações de entrada de instâncias na sub-rede privada devem redirecionar para as portas do Squid. O Squid escuta na porta 3129 para tráfego HTTP (em vez de 80) e na 3130 para tráfego HTTPS (em vez de 443). Use o comando iptables para rotear o tráfego:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

Criar um projeto do CodeBuild

Depois de configurar seu servidor proxy, você pode usá-lo AWS CodeBuild em uma sub-rede privada sem mais configurações. Cada solicitação HTTP e HTTPS passa pelo servidor de proxy público. Use o comando a seguir para visualizar o log de acesso do proxy do Squid:

```
sudo tail -f /var/log/squid/access.log
```

Executar um gerenciador de pacotes e outras ferramentas em um servidor de proxy

Use os procedimentos a seguir para executar um gerenciador de pacotes e outras ferramentas em um servidor proxy.

Como executar uma ferramenta, como um gerenciador de pacotes, em um servidor proxy

1. Adicione a ferramenta à lista de permissões em seu servidor de proxy incluindo instruções em seu arquivo `squid.conf`.
2. Adicione uma linha ao arquivo `buildspec` para que indique o endpoint privado do seu servidor de proxy.

Os exemplos a seguir demonstram como fazer isso para `apt-get`, `curl` e `maven`. Se você usar uma ferramenta diferente, os mesmos princípios são aplicáveis. Adicione-o a uma lista de

permissões no `squid.conf` arquivo e adicione um comando ao seu arquivo `buildspec` para CodeBuild informar o endpoint do seu servidor proxy.

Para executar **apt-get** em um servidor proxy

1. Adicione as instruções a seguir ao arquivo `squid.conf` para incluir `apt-get` na lista de permissões do seu servidor de proxy. As três primeiras linhas permitem que o `apt-get` seja executado no ambiente de compilação.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Adicione a instrução a seguir no seu arquivo `buildspec` para que os comandos `apt-get` procurem a configuração de proxy em `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

Para executar **curl** em um servidor proxy

1. Adicione o seguinte ao arquivo `squid.conf` para incluir `curl` em uma lista de permissões no ambiente de compilação.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a website. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Adicione a instrução a seguir no seu arquivo `buildspec` para que o `curl` use o servidor de proxy privado para acessar o site que você adicionou a `squid.conf`. Nesse exemplo, o site é `google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

Para executar **maven** em um servidor proxy

1. Adicione o seguinte ao arquivo `squid.conf` para incluir maven em uma lista de permissões no ambiente de compilação.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Adicione a seguinte instrução ao seu arquivo `buildspec`.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

AWS CloudFormation Modelo de VPC

AWS CloudFormation permite criar e provisionar implantações de AWS infraestrutura de forma previsível e repetida, usando arquivos de modelo para criar e excluir uma coleção de recursos juntos como uma única unidade (uma pilha). Para obter mais informações, consulte o [Guia do usuário do AWS CloudFormation](#).

Veja a seguir um modelo AWS CloudFormation YAML para configurar uma VPC a ser usada. AWS CodeBuild Esse arquivo também está disponível em [samples.zip](#).

```
Description: This template deploys a VPC, with a pair of public and private subnets
spread
across two Availability Zones. It deploys an internet gateway, with a default
route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),
and default routes for them in the private subnets.
```

Parameters:**EnvironmentName:**

Description: An environment name that is prefixed to resource names

Type: String

VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC

Type: String

Default: 10.192.0.0/16

PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

Type: String

Default: 10.192.10.0/24

PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

Type: String

Default: 10.192.11.0/24

PrivateSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 10.192.20.0/24

PrivateSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 10.192.21.0/24

Resources:**VPC:**

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCIDR

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: !Ref EnvironmentName

```
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: !Ref EnvironmentName

InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC

PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet1CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet2CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PrivateSubnet1CIDR
    MapPublicIpOnLaunch: false
    Tags:
```

- Key: Name
Value: !Sub \${EnvironmentName} Private Subnet (AZ1)

PrivateSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [1, !GetAZs '']

CidrBlock: !Ref PrivateSubnet2CIDR

MapPublicIpOnLaunch: false

Tags:

- Key: Name
Value: !Sub \${EnvironmentName} Private Subnet (AZ2)

NatGateway1EIP:

Type: AWS::EC2::EIP

DependsOn: InternetGatewayAttachment

Properties:

Domain: vpc

NatGateway2EIP:

Type: AWS::EC2::EIP

DependsOn: InternetGatewayAttachment

Properties:

Domain: vpc

NatGateway1:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt NatGateway1EIP.AllocationId

SubnetId: !Ref PublicSubnet1

NatGateway2:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt NatGateway2EIP.AllocationId

SubnetId: !Ref PublicSubnet2

PublicRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

```
Value: !Sub ${EnvironmentName} Public Routes
```

DefaultPublicRoute:

```
Type: AWS::EC2::Route
DependsOn: InternetGatewayAttachment
Properties:
  RouteTableId: !Ref PublicRouteTable
  DestinationCidrBlock: 0.0.0.0/0
  GatewayId: !Ref InternetGateway
```

PublicSubnet1RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PublicRouteTable
  SubnetId: !Ref PublicSubnet1
```

PublicSubnet2RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PublicRouteTable
  SubnetId: !Ref PublicSubnet2
```

PrivateRouteTable1:

```
Type: AWS::EC2::RouteTable
Properties:
  VpcId: !Ref VPC
  Tags:
    - Key: Name
      Value: !Sub ${EnvironmentName} Private Routes (AZ1)
```

DefaultPrivateRoute1:

```
Type: AWS::EC2::Route
Properties:
  RouteTableId: !Ref PrivateRouteTable1
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NatGateway1
```

PrivateSubnet1RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PrivateRouteTable1
  SubnetId: !Ref PrivateSubnet1
```



```
PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "no-ingress-sg"
    GroupDescription: "Security group with no ingress rule"
    VpcId: !Ref VPC

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ] ]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ] ]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1
```

PublicSubnet2:

Description: A reference to the public subnet in the 2nd Availability Zone

Value: !Ref PublicSubnet2

PrivateSubnet1:

Description: A reference to the private subnet in the 1st Availability Zone

Value: !Ref PrivateSubnet1

PrivateSubnet2:

Description: A reference to the private subnet in the 2nd Availability Zone

Value: !Ref PrivateSubnet2

NoIngressSecurityGroup:

Description: Security group with no ingress rule

Value: !Ref NoIngressSecurityGroup

Registro e monitoramento em AWS CodeBuild

O registro e o monitoramento são uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de AWS CodeBuild suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra. AWS fornece as seguintes ferramentas para monitorar seus CodeBuild recursos e construções e para responder a possíveis incidentes.

Tópicos

- [Registre chamadas de AWS CodeBuild API com AWS CloudTrail](#)
- [O monitor é CodeBuild construído com CloudWatch](#)

Registre chamadas de AWS CodeBuild API com AWS CloudTrail

AWS CodeBuild é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em CodeBuild. CloudTrail captura todas as chamadas de API para eventos CodeBuild as, incluindo chamadas do CodeBuild console e de chamadas de código para o. CodeBuild APIs Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos em um bucket do S3, incluindo eventos para CodeBuild. 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 CodeBuild, 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](#).

Tópicos

- [Sobre AWS CodeBuild as informações em CloudTrail](#)
- [Sobre as entradas do arquivo de AWS CodeBuild log](#)

Sobre AWS CodeBuild as informações em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre em CodeBuild, 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 eventos recentes em sua

AWS conta. Para obter mais informações, consulte [Visualização de CloudTrail eventos com histórico](#) de eventos no Guia AWS CloudTrail do usuário.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para CodeBuild, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do 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 S3 que você especificar. 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 suportados](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [Recebendo arquivos de CloudTrail log de várias contas](#)

Todas CodeBuild as ações são registradas CloudTrail e documentadas na [Referência da CodeBuild API](#). Por exemplo, chamadas para as ações CreateProject (no AWS CLI, create-project), StartBuild (no AWS CLI, start-project) e UpdateProject (no AWS CLI, update-project) geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do .
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o [elemento CloudTrail userIdentity](#) no Guia do AWS CloudTrail usuário.

Sobre as entradas do arquivo de AWS CodeBuild log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a

ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

Note

Para proteger informações confidenciais, o seguinte está oculto nos CodeBuild registros:

- AWS chave de acesso IDs. Para obter mais informações, consulte [Managing Access Keys for IAM Users](#) no Guia do usuário do AWS Identity and Access Management .
- Strings especificadas usando o repositório de parâmetros. Para obter mais informações, consulte [Systems Manager Parameter Store](#) e [Systems Manager Parameter Store Console Walkthrough](#) no Guia do usuário do Amazon EC2 Systems Manager.
- Cadeias de caracteres especificadas usando AWS Secrets Manager. Para obter mais informações, consulte [Gerenciamento de chaves](#).

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a criação de um projeto de compilação em CodeBuild.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "FederatedUser",
    "principalId": "account-ID:user-name",
    "arn": "arn:aws:sts::account-ID:federated-user/user-name",
    "accountId": "account-ID",
    "accessKeyId": "access-key-ID",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-09-06T17:59:10Z"
      },
      "sessionIssuer": {
        "type": "IAMUser",
        "principalId": "access-key-ID",
        "arn": "arn:aws:iam::account-ID:user/user-name",
        "accountId": "account-ID",
        "userName": "user-name"
      }
    }
  }
}
```

```

    },
    "eventTime": "2016-09-06T17:59:11Z",
    "eventSource": "codebuild.amazonaws.com",
    "eventName": "CreateProject",
    "awsRegion": "region-ID",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "user-agent",
    "requestParameters": {
      "awsActId": "account-ID"
    },
    },
    "responseElements": {
      "project": {
        "environment": {
          "image": "image-ID",
          "computeType": "BUILD_GENERAL1_SMALL",
          "type": "LINUX_CONTAINER",
          "environmentVariables": []
        },
        },
        "name": "codebuild-demo-project",
        "description": "This is my demo project",
        "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project:project-ID",
        "encryptionKey": "arn:aws:kms:region-ID:key-ID",
        "timeoutInMinutes": 10,
        "artifacts": {
          "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
          "type": "S3",
          "packaging": "ZIP",
          "outputName": "MyOutputArtifact.zip"
        },
        "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
        "lastModified": "Sep 6, 2016 10:59:11 AM",
        "source": {
          "type": "GITHUB",
          "location": "https://github.com/my-repo.git"
        },
        },
        "created": "Sep 6, 2016 10:59:11 AM"
      }
    },
    "requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
    "eventID": "581f7dd1-8d2e-40b0-aeee-0dbf7EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "account-ID"
  }
}

```

O monitor é CodeBuild construído com CloudWatch

Você pode usar CloudWatch a Amazon para observar suas construções, relatar quando algo está errado e realizar ações automáticas quando apropriado. Você pode monitorar as compilações em dois níveis:

Nível de projeto

Essas métricas servem para todas as compilações no projeto especificado. Para ver as métricas de um projeto, especifique o `ProjectName` para a dimensão no CloudWatch.

AWS nível da conta

Essas métricas servem para todas as compilações em uma conta. Para ver as métricas no nível da conta da AWS , não insira uma dimensão no CloudWatch. As métricas de utilização de recursos de criação não estão disponíveis no nível da AWS conta.

CloudWatch as métricas mostram o comportamento de suas construções ao longo do tempo. Por exemplo, é possível monitorar:

- Quantas construções foram tentadas em um projeto de compilação ou em uma AWS conta ao longo do tempo.
- Quantas construções foram bem-sucedidas em um projeto de construção ou em uma AWS conta ao longo do tempo.
- Quantas compilações falharam em um projeto de compilação ou em uma AWS conta ao longo do tempo.
- Quanto tempo CodeBuild gasto executando compilações em um projeto de compilação ou em uma AWS conta ao longo do tempo.
- Utilização de recursos de uma compilação ou um projeto de compilação inteiro. As métricas de utilização de recursos de compilação incluem métricas, como utilização de CPU, memória e armazenamento.

Para obter mais informações, consulte [Exibir CodeBuild métricas](#).

CodeBuild CloudWatch métricas

As métricas a seguir podem ser monitoradas por AWS conta ou por projeto de compilação. Para obter mais informações sobre como usar CloudWatch com CodeBuild, consulte [O monitor é CodeBuild construído com CloudWatch](#).

BuildDuration

Mede a duração da fase BUILD da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

Compilações

Mede o número de compilações acionadas.

Unidades: contagem

CloudWatch Estatísticas válidas: Soma

DownloadSourceDuration

Mede a duração da fase DOWNLOAD_SOURCE da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

Duração

Mede a duração de todas as compilações ao longo do tempo.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

FailedBuilds

Mede o número de compilações que falharam em virtude de um erro de cliente ou um tempo limite.

Unidades: contagem

CloudWatch Estatísticas válidas: Soma

FinalizingDuration

Mede a duração da fase FINALIZING da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

InstallDuration

Mede a duração da fase INSTALL da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

PostBuildDuration

Mede a duração da fase POST_BUILD da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

PreBuildDuration

Mede a duração da fase PRE_BUILD da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

ProvisioningDuration

Mede a duração da fase PROVISIONING da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

QueuedDuration

Mede a duração da fase QUEUED da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

SubmittedDuration

Mede a duração da fase SUBMITTED da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

SucceededBuilds

Mede o número de compilações bem-sucedidas.

Unidades: contagem

CloudWatch Estatísticas válidas: Soma

UploadArtifactsDuration

Mede a duração da fase UPLOAD_ARTIFACTS da compilação.

Unidades: segundos

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

CodeBuild CloudWatch métricas de utilização de recursos

Note

CodeBuild as métricas de utilização de recursos só estão disponíveis nas seguintes regiões:

- Região Ásia-Pacífico (Tóquio)
- Região Ásia-Pacífico (Seul)
- Região Ásia-Pacífico (Mumbai)
- Região Ásia-Pacífico (Singapura)
- Região Ásia-Pacífico (Sydney)
- Região Canadá (Central)
- Região Europa (Frankfurt)
- Região Europa (Irlanda)
- Região Europa (Londres)

- Região Europa (Paris)
- Região América do Sul (São Paulo)
- Região Leste dos EUA (N. da Virgínia)
- Região Leste dos EUA (Ohio)
- Região Oeste dos EUA (N. da Califórnia)
- Região Oeste dos EUA (Oregon)

As métricas de utilização de recursos a seguir podem ser monitoradas. Para obter mais informações sobre como usar CloudWatch com CodeBuild, consulte [O monitor é CodeBuild construído com CloudWatch](#).

CPUUtilized

O número de unidades de CPU do processamento alocado usado pelo contêiner de compilação.

Unidades: unidades de CPU

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

CPUUtilizedPorcentagem

A porcentagem de processamento alocado usado pelo contêiner de compilação.

Unidades: percentual

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

MemoryUtilized

O número de megabytes de memória usado pelo contêiner de compilação.

Unidades: megabytes

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

MemoryUtilizedPercent

A porcentagem de processamento alocado usado pelo contêiner de compilação.

Unidades: percentual

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

StorageReadBytes

A velocidade de leitura do armazenamento usada pelo contêiner de compilação.

Unidade: bytes/segundo

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

StorageWriteBytes

A velocidade de gravação do armazenamento usada pelo contêiner de compilação.

Unidade: bytes/segundo

CloudWatch Estatísticas válidas: Média (recomendada), Máxima, Mínima

CodeBuild CloudWatch dimensões

CodeBuild fornece as seguintes dimensões CloudWatch métricas. Se nenhuma delas for especificada, as métricas são para a AWS conta corrente.

BuildId, BuildNumber, ProjectName

As métricas são fornecidas para um identificador de compilação, número de compilação e nome do projeto.

ProjectName

As métricas são fornecidas para o nome de um projeto.

CodeBuild CloudWatch alarmes

Você pode usar o CloudWatch console para criar alarmes com base em CodeBuild métricas para poder reagir se algo der errado com suas compilações. As duas métricas mais úteis com alarmes são descritas nos itens a seguir. Para obter mais informações sobre como usar CloudWatch com CodeBuild, consulte [O monitor é CodeBuild construído com CloudWatch](#).

- **FailedBuild.** É possível criar um alarme que é acionado quando um número específico de compilações com falha é detectado em um número predefinido de segundos. Em CloudWatch, você especifica o número de segundos e quantas compilações com falha acionam um alarme.

- **Duration.** É possível criar um alarme que é acionado quando uma compilação leva mais tempo do que o esperado. Você especifica quantos segundos devem decorrer depois que uma compilação é iniciada e antes que uma compilação seja concluída, antes que o alarme seja acionado.

Para obter informações sobre como criar alarmes para CodeBuild métricas, consulte [Monitore CodeBuild construções com alarmes CloudWatch](#). Para obter mais informações sobre alarmes, consulte [Criação de CloudWatch alarmes da Amazon](#) no Guia CloudWatch do usuário da Amazon.

Exibir CodeBuild métricas

AWS CodeBuild monitora funções em seu nome e relata métricas por meio da Amazon CloudWatch. Essas métricas incluem o número total de compilações, compilações com falha, compilações bem-sucedidas e a duração de compilações.

Você pode usar o CodeBuild console ou o CloudWatch console para monitorar as métricas de CodeBuild. Os procedimentos a seguir mostram como exibir as métricas.

Tópicos

- [Exibir métricas de compilação \(CodeBuild console\)](#)
- [Exibir métricas de compilação \(CloudWatch console da Amazon\)](#)

Exibir métricas de compilação (CodeBuild console)

Note

Você não pode personalizar as métricas ou os gráficos usados para exibi-las no CodeBuild console. Se você quiser personalizar a exibição, use o CloudWatch console da Amazon para visualizar suas métricas de criação.

Métricas em nível de conta

Para exibir métricas no nível da conta da AWS

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. No painel de navegação, escolha Account metrics (Métricas da conta).

Métricas em nível de projeto

Para exibir métricas no nível do projeto

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Na lista de projetos de compilação, na coluna Name (Nome), escolha o projeto onde você deseja visualizar as métricas.
4. Escolha a guia Métricas.

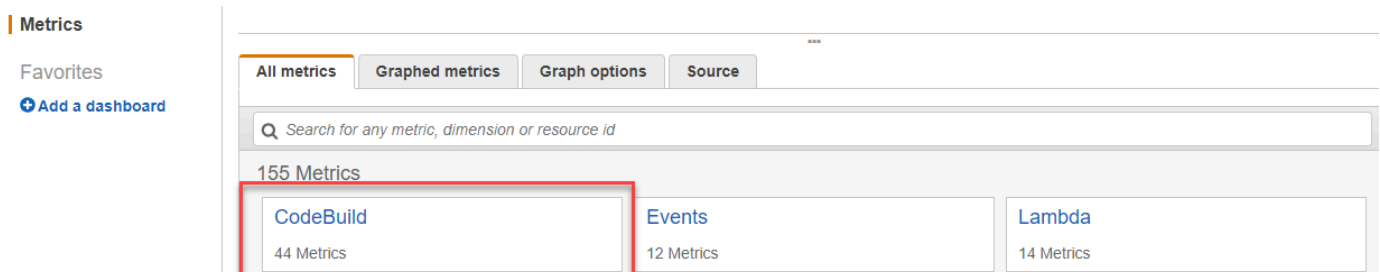
Exibir métricas de compilação (CloudWatch console da Amazon)

Você pode personalizar as métricas e os gráficos usados para exibi-las com o CloudWatch console.

Métricas em nível de conta

Para exibir métricas no nível da conta

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.
3. Na guia All metrics (Todas as métricas), escolha CodeBuild.

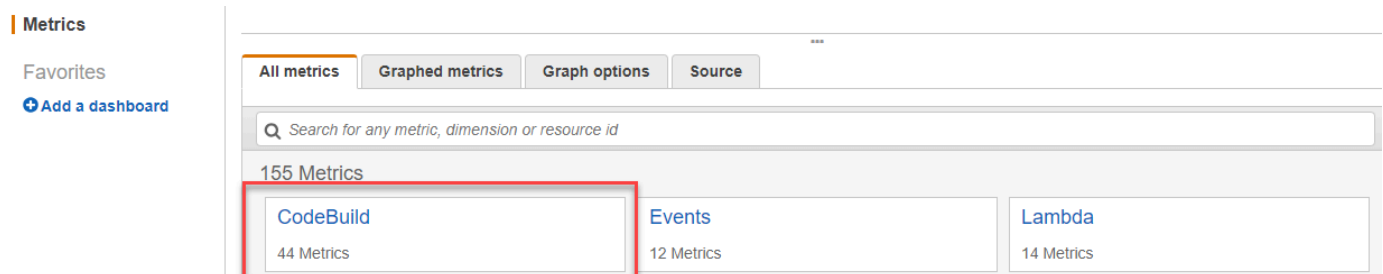


4. Escolha Métricas da conta.
5. Escolha um ou mais projetos e métricas. Para cada projeto, você pode escolher as métricas SucceededBuildsFailedBuilds,, Construções e Duração. Todas as combinações selecionadas de projeto e de métrica são exibidas no gráfico na página.

Métricas em nível de projeto

Para exibir métricas no nível do projeto

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.
3. Na guia All metrics (Todas as métricas), escolha CodeBuild.



4. Escolha Por projeto.
5. Escolha uma ou mais combinações de projeto e métrica. Para cada projeto, você pode escolher as métricas SucceededBuildsFailedBuilds,, Construções e Duração. Todas as combinações selecionadas de projeto e de métrica são exibidas no gráfico na página.
6. (Opcional) Você pode personalizar suas métricas e gráficos. Por exemplo, na lista suspensa na coluna Estatística, é possível escolher outra estatística para exibir. Ou, no menu suspenso na coluna Período, você pode escolher um período diferente a ser usado para monitorar as métricas.

Para obter mais informações, consulte [Métricas gráficas](#) e [Visualizar métricas disponíveis](#) no Guia CloudWatch do usuário da Amazon.

Exibir CodeBuild métricas de utilização de recursos

AWS CodeBuild monitora a utilização de recursos de construção em seu nome e relata métricas por meio da Amazon CloudWatch. Elas incluem métricas, como CPU, memória e utilização do armazenamento.

Note

CodeBuild as métricas de utilização de recursos são registradas somente para compilações que são executadas por mais de um minuto.

Você pode usar o CodeBuild console ou o CloudWatch console para monitorar as métricas de utilização de recursos para CodeBuild.

 Note

CodeBuild as métricas de utilização de recursos só estão disponíveis nas seguintes regiões:

- Região Ásia-Pacífico (Tóquio)
- Região Ásia-Pacífico (Seul)
- Região Ásia-Pacífico (Mumbai)
- Região Ásia-Pacífico (Singapura)
- Região Ásia-Pacífico (Sydney)
- Região Canadá (Central)
- Região Europa (Frankfurt)
- Região Europa (Irlanda)
- Região Europa (Londres)
- Região Europa (Paris)
- Região América do Sul (São Paulo)
- Região Leste dos EUA (N. da Virgínia)
- Região Leste dos EUA (Ohio)
- Região Oeste dos EUA (N. da Califórnia)
- Região Oeste dos EUA (Oregon)

Os procedimentos a seguir mostram como acessar as métricas de utilização de recursos.

Tópicos

- [Acesse métricas de utilização de recursos \(CodeBuild console\)](#)
- [Acesse métricas de utilização de recursos \(CloudWatch console da Amazon\)](#)

Acesse métricas de utilização de recursos (CodeBuild console)

Note

Você não pode personalizar as métricas ou os gráficos usados para exibi-las no CodeBuild console. Se você quiser personalizar a exibição, use o CloudWatch console da Amazon para visualizar suas métricas de criação.

Métricas de utilização de recursos em nível de projeto

Como acessar métricas de utilização de recursos em nível de projeto

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Projetos de compilação.
3. Na lista de projetos de compilação, na coluna Nome, escolha o projeto cujas métricas de utilização você deseja ver.
4. Escolha a guia Métricas. As métricas de utilização de recursos são exibidas na seção Métricas de utilização de recursos.
5. Para visualizar as métricas de utilização de recursos em nível de projeto no CloudWatch console, escolha Exibir CloudWatch na seção Métricas de utilização de recursos.

Métricas de utilização de recursos em nível de compilação

Como acessar métricas de utilização de recursos em nível de compilação

1. Faça login no AWS Management Console e abra o AWS CodeBuild console em <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. No painel de navegação, selecione Histórico de compilação.
3. Na lista de compilações, na coluna Execução da compilação, selecione a compilação cujas métricas de utilização você deseja visualizar.
4. Escolha a guia Utilização de recursos.
5. Para visualizar as métricas de utilização de recursos em nível de compilação no CloudWatch console, escolha Exibir CloudWatch na seção Métricas de utilização de recursos.

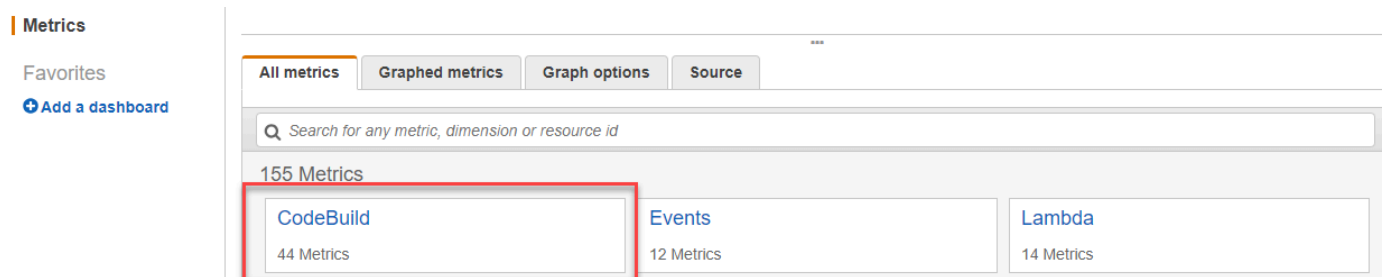
Acesse métricas de utilização de recursos (CloudWatch console da Amazon)

O CloudWatch console da Amazon pode ser usado para acessar métricas CodeBuild de utilização de recursos.

Métricas de utilização de recursos em nível de projeto

Como acessar métricas de utilização de recursos em nível de projeto

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.
3. Na guia All metrics (Todas as métricas), escolha CodeBuild.



4. Escolha Por projeto.
5. Escolha uma ou mais combinações de projetos e métricas para adicionar ao grafo. Todas as combinações selecionadas de projeto e de métrica são exibidas no gráfico na página.
6. (Opcional) É possível personalizar as métricas e os grafos na guia Métricas em gráficos. Por exemplo, na lista suspensa na coluna Estatística, é possível escolher outra estatística para exibir. Ou, no menu suspenso na coluna Período, você pode escolher um período diferente a ser usado para monitorar as métricas.

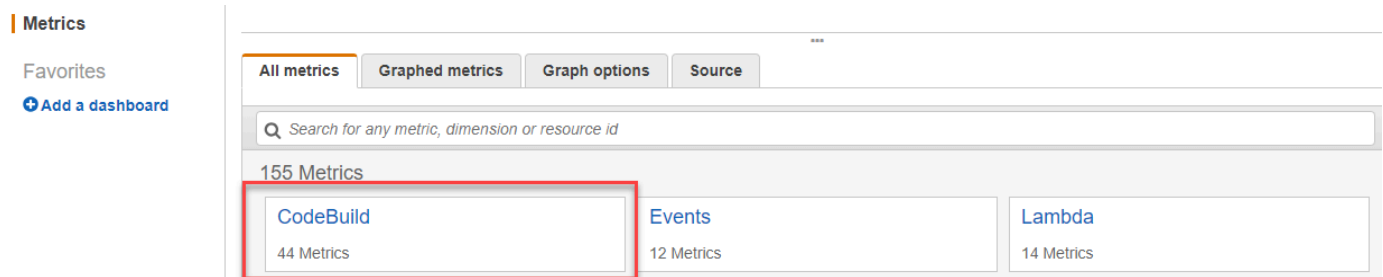
Para obter mais informações, consulte [Representação gráfica de métricas](#) e [Visualização de métricas disponíveis](#) no Guia do CloudWatch usuário da Amazon.

Métricas de utilização de recursos em nível de compilação

Como acessar métricas de utilização de recursos em nível de compilação

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.

3. Na guia All metrics (Todas as métricas), escolha CodeBuild.



4. Escolha BuildId, BuildNumber, ProjectName.

5. Selecione uma ou mais combinações de compilações e métricas para adicionar ao grafo. Todas as combinações selecionadas de compilação e de métrica são exibidas no grafo na página.
6. (Opcional) É possível personalizar as métricas e os grafos na guia Métricas em gráficos. Por exemplo, na lista suspensa na coluna Estatística, é possível escolher outra estatística para exibir. Ou, no menu suspenso na coluna Período, você pode escolher um período diferente a ser usado para monitorar as métricas.

Para obter mais informações, consulte [Representação gráfica de métricas](#) e [Visualização de métricas disponíveis](#) no Guia do CloudWatch usuário da Amazon.

Monitore CodeBuild construções com alarmes CloudWatch

Você pode criar um CloudWatch alarme para suas construções. Um alarme observa uma única métrica ao longo de um período especificado por você e realiza uma ou mais ações com base no valor da métrica em relação a um limite especificado ao longo de vários períodos. Usando a funcionalidade de CloudWatch alarme nativa, você pode especificar qualquer uma das ações suportadas CloudWatch quando um limite é excedido. Por exemplo, é possível especificar que uma notificação do Amazon SNS seja enviada quando mais de três compilações na conta falhar em 15 minutos.

Para criar um CloudWatch alarme para uma CodeBuild métrica

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Alarmes.
3. Escolha Create Alarm (Criar alarme).

4. Em CloudWatch Métricas por categoria, escolha CodeBuildMétricas. Se você souber que deseja apenas métricas no nível do projeto, escolha By Project (Por projeto). Se você souber que deseja apenas métricas no nível da conta, escolha Account Metrics (Métricas da conta).
5. Em Create Alarm (Criar alarme), se ainda não estiver selecionado, escolha Select Metric (Selecionar métrica).
6. Escolha uma métrica para a qual você deseja criar um alarme. As opções são By Project (Por projeto) ou Account Metrics (Métricas da conta).
7. Escolha Next (Próximo) ou Define Alarm (Definir alarme) e, em seguida, crie o alarme. Para obter mais informações, consulte [Criação de CloudWatch alarmes da Amazon](#) no Guia do CloudWatch usuário da Amazon. Para obter mais informações sobre como configurar notificações do Amazon SNS quando um alarme for acionado, consulte [Set up Amazon SNS notifications](#) no Guia do desenvolvedor do Amazon SNS.
8. Escolha Create Alarm.

Segurança em AWS CodeBuild

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

Segurança e conformidade são responsabilidades compartilhadas entre você AWS e você. Esse modelo compartilhado pode ajudar a aliviar sua carga operacional: AWS opera, gerencia e controla os componentes do sistema operacional host e da camada de virtualização até a segurança física das instalações de serviço. Você assume a responsabilidade e o gerenciamento do sistema operacional convidado (incluindo atualizações e patches de segurança) e outro software aplicativo associado. Você também é responsável pela configuração do firewall do grupo de segurança AWS fornecido. Suas responsabilidades variam de acordo com os serviços utilizados, a integração desses serviços ao seu ambiente de TI, bem como as leis e os regulamentos aplicáveis. Portanto, você deve considerar cuidadosamente os serviços que sua organização usa. Para obter mais informações, consulte o [Modelo de responsabilidade compartilhada](#).

Para saber como proteger seus CodeBuild recursos, consulte os tópicos a seguir.

Tópicos

- [Proteção de dados em AWS CodeBuild](#)
- [Gerenciamento de identidade e acesso em AWS CodeBuild](#)
- [Validação de conformidade para AWS CodeBuild](#)
- [Resiliência em AWS CodeBuild](#)
- [Segurança da infraestrutura em AWS CodeBuild](#)
- [Acesse seu provedor de origem em CodeBuild](#)
- [Prevenção contra o ataque do “substituto confuso” em todos os serviços](#)

Proteção de dados em AWS CodeBuild

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS CodeBuild. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a

privacidade de dados, consulte as [Data Privacy FAQ](#). Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and RGPD](#) no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com CodeBuild ou Serviços da AWS usa o console, a API ou AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Para proteger informações confidenciais, o seguinte está oculto nos CodeBuild registros:

- Cadeias de caracteres especificadas usando o Parameter Store nas variáveis de ambiente CodeBuild do projeto ou na seção `buildspecenv/parameter-store`. Para obter mais

informações, consulte o passo a [passo do console Systems Manager Parameter Store e Systems Manager Parameter Store](#) no Guia do usuário do Amazon EC2 Systems Manager.

- Cadeias de caracteres AWS Secrets Manager especificadas usando variáveis de ambiente CodeBuild do projeto ou a seção `buildspecenv/secrets-manager`. Para obter mais informações, consulte [Gerenciamento de chaves](#).

Para obter mais informações sobre proteção de dados, consulte a publicação [AWS modelo de responsabilidade compartilhada e GDPR](#) no blog de segurança da AWS .

Tópicos

- [Criptografia de dados](#)
- [Gerenciamento de chaves](#)
- [Privacidade do tráfego](#)

Criptografia de dados

A criptografia é uma parte importante da CodeBuild segurança. Algumas criptografias, como aquelas de dados em trânsito, são fornecidas por padrão e não requerem qualquer ação por parte do cliente. Outras criptografias, como aquelas de dados em repouso, podem ser configuradas durante a criação do projeto ou compilação.

- Criptografia de dados em repouso - artefatos de criação, como cache, registros, arquivos de dados brutos de relatórios de teste exportados e resultados de compilação, são criptografados por padrão usando Chaves gerenciadas pela AWS. Se não quiser usar essas chaves do KMS, deverá criar e configurar uma chave gerenciada pelo cliente. Para obter mais informações, consulte [Criar chaves do KMS](#) e [Conceitos do AWS Key Management Service](#) no Guia do usuário do AWS Key Management Service .
- Você pode armazenar o identificador da chave AWS KMS CodeBuild usada para criptografar o artefato de saída da compilação na `CODEBUILD_KMS_KEY_ID` variável de ambiente. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).
- É possível especificar uma chave gerenciada pelo cliente ao criar um projeto de compilação. Para obter mais informações, consulte [Set the Encryption Key Using the Console](#) e [Definir a chave de criptografia usando a CLI](#).

Os volumes do Amazon Elastic Block Store da sua frota de construção são criptografados por padrão usando Chaves gerenciadas pela AWS.

- Criptografia de dados em trânsito - Toda a comunicação entre clientes CodeBuild CodeBuild e entre suas dependências posteriores é protegida usando conexões TLS assinadas usando o processo de assinatura Signature Version 4. Todos os CodeBuild endpoints usam certificados SHA-256 que são gerenciados pelo AWS Private Certificate Authority Para obter mais informações, consulte [Processo de assinatura do Signature versão 4](#) e [O que é o ACM PCA?](#).
- Criptografia de artefatos de compilação - A função de CodeBuild serviço associada ao projeto de compilação requer acesso a uma chave KMS para criptografar seus artefatos de saída de compilação. Por padrão, CodeBuild usa um Chave gerenciada pela AWS para o Amazon S3 em sua AWS conta. Se você não quiser usar essa Chave gerenciada pela AWS, deverá criar e configurar uma chave gerenciada pelo cliente. Para obter mais informações, consulte [Criptografar saídas da compilação](#) e [Creating keys](#) no Guia do desenvolvedor do AWS KMS .

Gerenciamento de chaves

Você pode proteger seu conteúdo contra o uso não autorizado. Armazene suas chaves de criptografia e AWS Secrets Manager, em seguida, conceda à função de CodeBuild serviço associada ao projeto de construção permissão para obter as chaves de criptografia da sua conta do Secrets Manager. Para obter mais informações, consulte [Criptografar saídas da compilação usando uma chave gerenciada pelo cliente](#), [Crie um projeto de construção em AWS CodeBuild](#), [Execute AWS CodeBuild compilações manualmente](#) e [Tutorial: Armazenar e recuperar um segredo](#).

Use a variável de CODEBUILD_KMS_KEY_ID ambiente em um comando de construção para obter o identificador da AWS KMS chave. Para obter mais informações, consulte [Variáveis de ambiente em ambientes de compilação](#).

É possível usar o Secrets Manager para proteger as credenciais para um registro privado que armazene uma imagem do Docker usada para o ambiente de runtime. Para obter mais informações, consulte [Registro privado com AWS Secrets Manager amostra para CodeBuild](#).

Privacidade do tráfego

Você pode melhorar a segurança de suas compilações configurando CodeBuild para usar uma interface VPC endpoint. Para fazer isso, você não precisa de um gateway da Internet, de um dispositivo NAT ou de um gateway privado virtual. Também não é necessário configurá-lo PrivateLink, embora seja recomendado. Para obter mais informações, consulte [Usar VPC endpoints](#). Para obter mais informações sobre endpoints de VPC PrivateLink e VPC, consulte [Acessando AWS serviços por AWS PrivateLink](#) meio de. PrivateLink

Gerenciamento de identidade e acesso em AWS CodeBuild

O acesso a AWS CodeBuild requer credenciais. Essas credenciais devem ter permissões para acessar AWS recursos, como armazenar e recuperar artefatos de construção em buckets do S3 e visualizar Amazon CloudWatch Logs para compilações. As seções a seguir descrevem como você pode usar o [AWS Identity and Access Management](#)(IAM) e como ajudar CodeBuild a proteger o acesso aos seus recursos:

Visão geral do gerenciamento de permissões de acesso aos seus AWS CodeBuild recursos

Cada AWS recurso pertence a uma AWS conta, e 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).

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 receberá as permissões, os recursos que poderão acessar e as ações que poderão ser executadas nesses recursos.

Tópicos

- [AWS CodeBuild recursos e operações](#)
- [Informações sobre propriedade de recursos](#)
- [Gerenciar acesso aos recursos da](#)
- [Especificar elementos da política: ações, efeitos e entidades principais](#)

AWS CodeBuild recursos e operações

Em AWS CodeBuild, o recurso principal é um projeto de construção. Em uma política, você usa um Amazon Resource Name (ARN) para identificar o recurso a que a política se aplica. As construções também são recursos e estão ARNs associadas a elas. Para obter mais informações, consulte

[Amazon Resource Names \(ARN\) e AWS Service Namespaces](#) no. Referência geral da Amazon Web Services

Tipo de recurso	Formato ARN
Projeto de compilação	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :project/<i>project-name</i></code>
Compilar	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :build/<i>build-ID</i></code>
Grupo de relatórios	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report-group/<i>report-group-name</i></code>
Relatório	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report/<i>report-ID</i></code>
Frota	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :fleet/<i>fleet-ID</i></code>
Todos os CodeBuild recursos	<code>arn:aws:codebuild:*</code>
Todos os CodeBuild recursos pertencentes à conta especificada na AWS região especificada	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :*</code>

Important

Ao usar o recurso de capacidade reservada, os dados armazenados em cache nas instâncias da frota, incluindo arquivos de origem, camadas do Docker e diretórios em cache especificados no buildspec, podem ser acessados por outros projetos na mesma conta. Isso ocorre por design e permite que projetos dentro da mesma conta compartilhem instâncias de frota.

 Note

A maioria dos AWS serviços trata dois pontos (:) ou uma barra invertida (/) como o mesmo caractere em ARNs. No entanto, CodeBuild usa uma correspondência exata nos padrões e regras dos recursos. Certifique-se de usar os caracteres corretos ao criar padrões de evento, de modo que eles correspondam à sintaxe ARN no recurso.

Por exemplo, você pode indicar um projeto de construção específico (*myBuildProject*) em sua declaração usando seu ARN da seguinte forma:

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Para especificar todos os recursos, ou se uma ação de API não for compatível com ARNs, use o caractere curinga (*) no Resource elemento da seguinte forma:

```
"Resource": "*"
```

Algumas ações CodeBuild da API aceitam vários recursos (por exemplo, `BatchGetProjects`). Para especificar vários recursos em uma única instrução, separe-os ARNs com vírgulas, da seguinte forma:

```
"Resource": [  
  "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",  
  "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"  
]
```

CodeBuild fornece um conjunto de operações para trabalhar com os CodeBuild recursos. Para ver uma lista, consulte [AWS CodeBuild referência de permissões](#).

Informações sobre propriedade de recursos

A AWS conta é proprietária dos recursos criados na conta, independentemente de quem criou os recursos. Especificamente, o proprietário do recurso é a AWS conta da [entidade principal](#) (ou seja, a conta raiz, um usuário ou uma função do IAM) que autentica a solicitação de criação do recurso. Os seguintes exemplos mostram como isso funciona:

- Se você usar as credenciais da conta raiz da sua AWS conta para criar uma regra, sua AWS conta é a proprietária do CodeBuild recurso.

- Se você criar um usuário em sua AWS conta e conceder permissões para criar CodeBuild recursos para esse usuário, o usuário poderá criar CodeBuild recursos. No entanto, sua AWS conta, à qual o usuário pertence, possui os CodeBuild recursos.
- Se você criar uma função do IAM em sua AWS conta com permissões para criar CodeBuild recursos, qualquer pessoa que possa assumir a função poderá criar CodeBuild recursos. Sua AWS conta, à qual a função pertence, é proprietária dos CodeBuild recursos.

Gerenciar acesso aos recursos da

A política de permissões descreve quem possui acesso a quais recursos.

Note

Esta seção aborda o uso do IAM no AWS CodeBuild. Não são fornecidas informações detalhadas sobre o serviço IAM. Para obter a documentação completa do IAM, consulte [O que é o IAM?](#) no Guia do usuário do IAM. Para obter mais informações sobre a sintaxe e as descrições da política do IAM, consulte a [Referência de políticas do AWS IAM](#) no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM). As políticas anexadas a um recurso são chamadas de políticas baseadas em recursos. CodeBuild suporta políticas baseadas em identidade e políticas baseadas em recursos para determinadas leituras somente APIs para fins de compartilhamento de recursos entre contas.

Acesso seguro aos buckets do S3

É altamente recomendável que você inclua as seguintes permissões em sua função do IAM para verificar se o bucket do S3 associado ao seu CodeBuild projeto pertence a você ou a alguém em quem você confia. Essas permissões não estão incluídas nas políticas e funções AWS gerenciadas. Você mesmo deve adicioná-las.

- `s3:GetBucketAcl`
- `s3:GetBucketLocation`

Se o proprietário de um bucket do S3 usado no projeto for alterado, você deverá verificar se o bucket ainda é de sua propriedade e, caso contrário, atualizar o perfil do IAM. Para obter mais informações,

consulte [Permita que os usuários interajam com CodeBuild](#) e [CodeBuild Permitir interagir com outros AWS serviços](#).

Especificar elementos da política: ações, efeitos e entidades principais

Para cada AWS CodeBuild recurso, o serviço define um conjunto de operações de API. Para conceder permissões para essas operações de API, CodeBuild defina um conjunto de ações que você pode especificar em uma política. Algumas operações da API podem exigir permissões para mais de uma ação a fim de realizar a operação da API. Para obter mais informações, consulte [AWS CodeBuild recursos e operações](#) e [AWS CodeBuild referência de permissões](#).

Estes são os elementos de política básicos:

- **Recurso:** use um nome de recurso da Amazon (ARN) para identificar o recurso ao qual a política se aplica.
- **Ação:** use palavras-chave de ações para identificar operações de recursos que deseja permitir ou negar. Por exemplo, a permissão `codebuild:CreateProject` permite que o usuário execute a operação `CreateProject`.
- **Efeito:** especifique o efeito, permitir ou negar, quando o usuário solicita a ação. Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar acesso explicitamente a um recurso. Você poderia fazer isso para garantir que um usuário não possa acessar um recurso, mesmo quando uma política diferente concede o acesso.
- **Entidade principal:** em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é a entidade principal implícita. Para as políticas baseadas em recursos, você especifica o usuário, conta, serviço ou outra entidade a receber permissões.

Para saber mais sobre a sintaxe e as descrições da política do IAM, consulte a [Referência de política do AWS IAM](#) no Guia do usuário do IAM.

Para ver uma tabela mostrando todas as ações da CodeBuild API e os recursos aos quais elas se aplicam, consulte [AWS CodeBuild referência de permissões](#) o.

Usando políticas baseadas em identidade para AWS CodeBuild

Este tópico fornece exemplos de políticas baseadas em identidade que demonstram como um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários,

grupos e funções) e, assim, conceder permissões para realizar operações em recursos do AWS CodeBuild .

Important

Recomendamos que você primeiro analise os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos seus CodeBuild recursos. Para obter mais informações, consulte [Visão geral do gerenciamento de permissões de acesso aos seus AWS CodeBuild recursos](#).

Tópicos

- [Permissões necessárias para usar o console do AWS CodeBuild](#)
- [Permissões necessárias para se conectar AWS CodeBuild ao Amazon Elastic Container Registry](#)
- [Permissões necessárias para que o AWS CodeBuild console se conecte aos provedores de origem](#)
- [AWS políticas gerenciadas \(predefinidas\) para AWS CodeBuild](#)
- [CodeBuild políticas e notificações gerenciadas](#)
- [CodeBuild atualizações nas políticas AWS gerenciadas](#)
- [Exemplos de política gerenciada pelo cliente](#)

O exemplo a seguir mostra uma política de permissões que permite que um usuário obtenha informações sobre projetos de build somente na região us-east-2, para a conta 123456789012, para qualquer projeto de build que comece com o nome my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permissões necessárias para usar o console do AWS CodeBuild

Um usuário que usa o AWS CodeBuild console deve ter um conjunto mínimo de permissões que permita ao usuário descrever outros AWS recursos para a AWS conta. Você deve ter permissões dos seguintes serviços:

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (se você estiver armazenando seu código-fonte em um AWS CodeCommit repositório)
- Amazon Elastic Container Registry (Amazon ECR) (se você estiver usando um ambiente de compilação que dependa de uma imagem do Docker em um repositório do Amazon ECR)

Note

Desde 26 de julho de 2022, a política padrão do IAM vem sendo atualizada. Para obter mais informações, consulte [Permissões necessárias para se conectar AWS CodeBuild ao Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (se você estiver usando um ambiente de compilação que dependa de uma imagem do Docker em um repositório do Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Se você criar uma política do IAM mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido.

Permissões necessárias para se conectar AWS CodeBuild ao Amazon Elastic Container Registry

Em 26 de julho de 2022, AWS CodeBuild atualizou sua política padrão do IAM para a permissão Amazon ECR. As seguintes permissões foram removidas da política padrão:

```
"ecr:PutImage",  
"ecr:InitiateLayerUpload",  
"ecr:UploadLayerPart",
```

```
"ecr:CompleteLayerUpload"
```

Para CodeBuild projetos criados antes de 26 de julho de 2022, recomendamos que você atualize sua política com a seguinte política do Amazon ECR:

```
"Action": [  
    "ecr:BatchCheckLayerAvailability",  
    "ecr:GetDownloadUrlForLayer",  
    "ecr:BatchGetImage"  
]
```

Para obter mais informações sobre a atualização da política, consulte [Permita que os usuários interajam com CodeBuild](#).

Permissões necessárias para que o AWS CodeBuild console se conecte aos provedores de origem

O AWS CodeBuild console usa as seguintes ações de API para se conectar aos provedores de origem (por exemplo, GitHub repositórios).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Você pode associar provedores de origem (como GitHub repositórios) aos seus projetos de compilação usando o AWS CodeBuild console. Para fazer isso, primeiro você deve adicionar as ações de API anteriores às políticas de acesso do IAM associadas ao usuário que você usa para acessar o AWS CodeBuild console.

As ações de API `ListConnectedOAuthAccounts`, `ListRepositories` e `PersistOAuthToken` não devem ser chamadas pelo código. Portanto, essas ações de API não estão incluídas no AWS CLI AWS SDKs e.

AWS políticas gerenciadas (predefinidas) para AWS CodeBuild

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, para que você não precise investigar quais permissões

são necessárias. As políticas gerenciadas CodeBuild também fornecem permissões para realizar operações em outros serviços, como IAM, Amazon AWS CodeCommit EC2, Amazon ECR, Amazon SNS e CloudWatch Amazon Events, conforme exigido para as responsabilidades dos usuários que receberam a política em questão. Por exemplo, a `AWSCodeBuildAdminAccess` política é uma política de usuário de nível administrativo que permite aos usuários com essa política criar e gerenciar regras de CloudWatch eventos para compilações de projetos e tópicos do Amazon SNS para notificações sobre eventos relacionados a projetos (tópicos cujos nomes são prefixados com `arn:aws:codebuild:`), bem como administrar projetos e grupos de relatórios em CodeBuild. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Guia do usuário do IAM.

As políticas AWS gerenciadas a seguir, que você pode anexar aos usuários em sua conta, são específicas de AWS CodeBuild.

`AWSCodeBuildAdminAccess`

Fornecer acesso total à CodeBuild inclusão de permissões para administrar projetos de CodeBuild compilação.

`AWSCodeBuildDeveloperAccess`

Fornecer acesso CodeBuild , mas não permite, a administração do projeto de compilação.

`AWSCodeBuildReadOnlyAccess`

Fornecer acesso somente para leitura a CodeBuild

Para acessar os artefatos de saída de compilação CodeBuild criados, você também deve anexar a política AWS gerenciada chamada `AmazonS3ReadOnlyAccess`.

Para criar e gerenciar funções CodeBuild de serviço, você também deve anexar a política AWS gerenciada chamada `IAMFullAccess`.

Você também pode criar as próprias políticas do IAM personalizadas a fim de conceder permissões para ações e recursos do CodeBuild. Você pode anexar essas políticas personalizadas a usuários ou grupos do que exijam essas permissões.

Tópicos

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

AWSCodeBuildAdminAccess

A `AWSCodeBuildAdminAccess` política fornece acesso total a CodeBuild, incluindo permissões para administrar projetos de CodeBuild compilação. Aplique essa política somente a usuários de nível administrativo para conceder a eles controle total sobre CodeBuild projetos, grupos de relatórios e recursos relacionados em sua AWS conta, incluindo a capacidade de excluir projetos e grupos de relatórios.

A política `AWSCodeBuildAdminAccess` contém a seguinte declaração:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSServicesAccess",
      "Action": [
        "codebuild:*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "codecommit:ListRepositories",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeVpcs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "elasticfilesystem:DescribeFileSystems",
        "events>DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Effect": "Allow",
    }
  ]
}
```

```
    "Resource": "*"
  },
  {
    "Sid": "CWLDeleteLogGroupAccess",
    "Action": [
      "logs:DeleteLogGroup"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:logs:*:*:log-group:/aws/codebuild/*:log-stream:*"
  },
  {
    "Sid": "SSMParameterWriteAccess",
    "Effect": "Allow",
    "Action": [
      "ssm:PutParameter"
    ],
    "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
  },
  {
    "Sid": "SSMStartSessionAccess",
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession"
    ],
    "Resource": "arn:aws:ecs:*:*:task/*/*"
  },
  {
    "Sid": "CodeStarConnectionsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-connections:CreateConnection",
      "codestar-connections>DeleteConnection",
      "codestar-connections:UpdateConnectionInstallation",
      "codestar-connections:TagResource",
      "codestar-connections:UntagResource",
      "codestar-connections:ListConnections",
      "codestar-connections:ListInstallationTargets",
      "codestar-connections:ListTagsForResource",
      "codestar-connections:GetConnection",
      "codestar-connections:GetIndividualAccessToken",
      "codestar-connections:GetInstallationUrl",
      "codestar-connections:PassConnection",
      "codestar-connections:StartOAuthHandshake",
      "codestar-connections:UseConnection"
    ]
  }
}
```

```

    ],
    "Resource": [
        "arn:aws:codestar-connections:*:*:connection/*",
        "arn:aws:codeconnections:*:*:*"
    ]
},
{
    "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:codebuild:*:*:project/*"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
    "Effect": "Allow",
    "Action": [
        "sns:CreateTopic",
        "sns:SetTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:codestar-notifications*"
}

```

```
    },
    {
      "Sid": "SNSTopicListAccess",
      "Effect": "Allow",
      "Action": [
        "sns:ListTopics",
        "sns:GetTopicAttributes"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeStarNotificationsChatbotAccess",
      "Effect": "Allow",
      "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
      ],
      "Resource": "*"
    }
  ]
}
```

AWSCodeBuildDeveloperAccess

A `AWSCodeBuildDeveloperAccess` política permite acesso a todas as funcionalidades e recursos relacionados ao projeto CodeBuild e ao grupo de relatórios. Essa política não permite que os usuários excluam CodeBuild projetos, grupos de relatórios ou recursos relacionados em outros AWS serviços, como CloudWatch Eventos. Recomendamos que você aplique essa política à maioria dos usuários.

A política `AWSCodeBuildDeveloperAccess` contém a seguinte declaração:

```
{
  "Statement": [
    {
      "Sid": "AWSServicesAccess",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:StartBuildBatch",
        "codebuild:StopBuildBatch",
        "codebuild:RetryBuild",
        "codebuild:RetryBuildBatch",
```

```

        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
    ],
    "Effect": "Allow",
    "Resource": "*"
},
{
    "Sid": "SSMParameterWriteAccess",
    "Effect": "Allow",
    "Action": [
        "ssm:PutParameter"
    ],
    "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
},
{
    "Sid": "SSMStartSessionAccess",
    "Effect": "Allow",
    "Action": [
        "ssm:StartSession"
    ],
    "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
    "Sid": "CodeStarConnectionsUserAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
    ],
    "Resource": [

```

```

        "arn:aws:codestar-connections:*:*:connection/*",
        "arn:aws:codeconnections:*:*:*"
    ]
},
{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "codestar-notifications:NotificationsForResource":
"arn:aws:codebuild:*:*:project/*"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics",
        "sns:GetTopicAttributes"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",

```

```
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }
],
"Version": "2012-10-17"
}
```

AWSCodeBuildReadOnlyAccess

A `AWSCodeBuildReadOnlyAccess` política concede acesso somente para leitura CodeBuild e recursos relacionados em outros AWS serviços. Aplique esta política a usuários que possam visualizar e executar compilações, visualizar projetos e grupos de relatórios, mas que não possam fazer alterações neles.

A política `AWSCodeBuildReadOnlyAccess` contém a seguinte declaração:

```
{
  "Statement": [
    {
      "Sid": "AWSServicesAccess",
      "Action": [
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:List*",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
  ],
}
```



```
    "Sid": "CodeStarConnectionsUserAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
    ],
    "Resource": [
        "arn:aws:codestar-connections:*:*:connection/*",
        "arn:aws:codeconnections:*:*:*"
    ]
},
{
    "Sid": "CodeStarNotificationsPowerUserAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:DescribeNotificationRule"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "codestar-notifications:NotificationsForResource":
"arn:aws:codebuild:*:*:project/*"
        }
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets"
    ],
    "Resource": "*"
}
],
"Version": "2012-10-17"
}
```

CodeBuild políticas e notificações gerenciadas

CodeBuild suporta notificações, que podem notificar os usuários sobre mudanças importantes na criação de projetos. Políticas gerenciadas para CodeBuild incluem declarações de política para funcionalidade de notificação. Para obter mais informações, consulte [O que são notificações?](#).

Permissões relacionadas a notificações em políticas gerenciadas somente leitura

A política gerenciada `AWSCodeBuildReadOnlyAccess` 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": {
    "ArnLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
  }
},
{
  "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 `AWSCodeBuildDeveloperAccess` 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:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition" : {
        "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}

```

Para obter mais informações sobre o IAM e notificações, consulte [Identity and Access Management for AWS CodeStar Notifications](#).

CodeBuild atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas CodeBuild desde que esse serviço começou a rastrear essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, assine o feed RSS em [AWS CodeBuild Histórico do documento do Guia do Usuário](#).

Alteração	Descrição	Data
AWSCodeBuildAdminAccess , AWSCodeBuildDeveloperAccess e AWSCodeBuildReadOnlyAccess : atualização de políticas existentes	CodeBuild atualizou um recurso para essas políticas. As AWSCodeBuildReadOnlyAccess políticas AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess , e foram alteradas para atualizar um recurso existente. O recurso original arn:aws:codebuild:* foi atualizado paraarn:aws:codebuild:*:*:project/* .	15 de novembro de 2024
AWSCodeBuildAdminAccess , AWSCodeBuildDeveloperAccess e AWSCodeBuildReadOnlyAccess : atualização de políticas existentes	CodeBuild adicionou um recurso a essas políticas para apoiar o Conexões de código da AWS rebranding. As políticas AWSCodeBuildAdminAccess , AWSCodeBuildDeveloperAccess e AWSCodeBuildReadOnlyAccess foram alteradas para adicionar um recurso, arn:aws:c	18 de abril de 2024

Alteração	Descrição	Data
	<pre>codeconnections:*:* :* .</pre>	
AWSCodeBuildAdminAccess e AWSCodeBuildDeveloperAccess : atualização em políticas existentes	CodeBuild 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 políticas AWSCodeBuildAdminAccess e AWSCodeBuildDeveloperAccess foram alteradas para adicionar uma permissão, chatbot:ListMicrosoftTeamsChannelConfigurations .	16 de maio de 2023
CodeBuild começou a rastrear alterações	CodeBuild começou a rastrear as mudanças em suas políticas AWS gerenciadas.	16 de maio de 2021

Exemplos de política gerenciada pelo cliente

Nesta seção, você pode encontrar exemplos de políticas de usuário que concedem permissões para ações do AWS CodeBuild . Essas políticas funcionam quando você está usando a CodeBuild API, AWS SDKs, ou AWS CLI. Ao usar o console, é necessário conceder permissões adicionais específicas e ele. Para mais informações, consulte [Permissões necessárias para usar o console do AWS CodeBuild](#).

Você pode usar os exemplos de políticas do IAM a seguir para limitar o CodeBuild acesso de seus usuários e funções.

Tópicos

- [Permitir que um usuário obtenha informações sobre projetos de compilação](#)
- [Permitir que um usuário obtenha informações sobre frotas](#)
- [Permitir que um usuário obtenha informações sobre grupos de relatórios](#)
- [Permitir que um usuário obtenha informações sobre relatórios](#)
- [Permitir que um usuário crie projetos de compilação](#)
- [Permitir que um usuário crie uma frota](#)
- [Permitir que um usuário crie um grupo de relatórios](#)
- [Permitir que um usuário exclua uma frota](#)
- [Permitir que um usuário exclua um grupo de relatórios](#)
- [Permitir que um usuário exclua um relatório](#)
- [Permitir que um usuário exclua projetos de compilação](#)
- [Permitir que um usuário obtenha uma lista de nomes de projetos de compilação](#)
- [Permitir que um usuário altere informações sobre projetos de compilação](#)
- [Permitir que um usuário altere uma frota](#)
- [Permitir que um usuário altere um grupo de relatórios](#)
- [Permitir que um usuário obtenha informações sobre compilações](#)
- [Permitir que um usuário obtenha uma lista de compilações IDs para um projeto de construção](#)
- [Permitir que um usuário obtenha uma lista de construções IDs](#)
- [Permitir que um usuário obtenha uma lista de frotas](#)
- [Permitir que um usuário obtenha uma lista de grupos de relatórios](#)
- [Permitir que um usuário obtenha uma lista de relatórios](#)
- [Permitir que um usuário obtenha uma lista de relatórios para um grupo de relatórios](#)
- [Permitir que um usuário obtenha uma lista de casos de teste para um relatório](#)
- [Permitir que um usuário comece a executar compilações](#)
- [Permitir que um usuário tente parar compilações](#)
- [Permitir que um usuário tente excluir compilações](#)
- [Permitir que um usuário obtenha informações sobre imagens do Docker que são gerenciadas pelo CodeBuild](#)
- [Permitir que um usuário adicione uma política de permissão para um perfil de serviço de frota](#)

- [Permitir CodeBuild acesso aos AWS serviços necessários para criar uma interface de rede VPC](#)
- [Use uma declaração de negação para AWS CodeBuild evitar a desconexão dos provedores de origem](#)

Permitir que um usuário obtenha informações sobre projetos de compilação

O seguinte exemplo de declaração de política permite que um usuário obtenha informações sobre projetos de compilação na região us-east-2, para a conta 123456789012, para qualquer projeto de compilação que comece com o nome my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que um usuário obtenha informações sobre frotas

O seguinte exemplo de declaração de políticas permite que um usuário obtenha informações sobre frotas na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetFleets",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:fleet/*"
    }
  ]
}
```

Permitir que um usuário obtenha informações sobre grupos de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário obtenha informações sobre grupos de relatórios na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário obtenha informações sobre relatórios

O seguinte exemplo de declaração de políticas permite que um usuário obtenha informações sobre relatórios na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário crie projetos de compilação

O exemplo de declaração de política a seguir permite que um usuário crie projetos de construção com qualquer nome, mas somente na us-east-2 Região para a conta 123456789012 e usando somente a função CodeBuild de serviço especificada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
```



```
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
  }
]
```

O exemplo de declaração de política a seguir permite que um usuário crie projetos de construção com qualquer nome, mas somente na us-east-2 Região para a conta 123456789012 e usando somente a função CodeBuild de serviço especificada. Também impõe que o usuário só possa usar a função de serviço especificada com AWS CodeBuild e não com nenhum outro AWS serviço.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole",
      "Condition": {
        "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
      }
    }
  ]
}
```

Permitir que um usuário crie uma frota

O seguinte exemplo de declaração de políticas permite que um usuário crie uma frota na região us-east-2, para a conta 123456789012:

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

```
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:fleet/*"
  }
]
}
```

Permitir que um usuário crie um grupo de relatórios

O exemplo de declaração de política a seguir permite que um usuário crie um grupo de relatórios na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário exclua uma frota

O seguinte exemplo de declaração de políticas permite que um usuário exclua uma frota na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteFleet",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:fleet/*"
    }
  ]
}
```

Permitir que um usuário exclua um grupo de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário exclua um grupo de relatórios na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário exclua um relatório

O seguinte exemplo de declaração de políticas permite que um usuário exclua um relatório na região `us-east-2`, para a conta `123456789012`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReport",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário exclua projetos de compilação

O seguinte exemplo de declaração de política permite que um usuário exclua projetos de compilação na região `us-east-2`, para a conta `123456789012`, para qualquer projeto de compilação que comece com o nome `my`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

```
]
}
```

Permitir que um usuário obtenha uma lista de nomes de projetos de compilação

O seguinte exemplo de declaração de política permite que um usuário obtenha uma lista de nomes de projeto de build para a mesma conta:

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

Permitir que um usuário altere informações sobre projetos de compilação

O seguinte exemplo de declaração de política permite que um usuário altere informações sobre projetos de compilação com qualquer nome, porém somente na região us-east-2, para a conta 123456789012, e somente para a função de serviço AWS CodeBuild especificada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
    }
  ]
}
```

Permitir que um usuário altere uma frota

O seguinte exemplo de declaração de políticas permite que um usuário altere uma frota na região `us-east-2`, para a conta `123456789012`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:fleet/*"
    }
  ]
}
```

Permitir que um usuário altere um grupo de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário altere um grupo de relatórios na região `us-east-2`, para a conta `123456789012`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário obtenha informações sobre compilações

O seguinte exemplo de declaração de políticas permite que um usuário obtenha informações sobre compilações na região `us-east-2`, para a conta `123456789012`, para projetos de compilação denominados `my-build-project` e `my-other-build-project`:

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

```
{
  "Effect": "Allow",
  "Action": "codebuild:BatchGetBuilds",
  "Resource": [
    "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
    "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
  ]
}
```

Permitir que um usuário obtenha uma lista de compilações IDs para um projeto de construção

O exemplo de declaração de política a seguir permite que um usuário obtenha uma lista de construções IDs na us-east-2 região 123456789012 para considerar os projetos de construção chamados my-build-project e my-other-build-project:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
      "Resource": [
        "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
        "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
      ]
    }
  ]
}
```

Permitir que um usuário obtenha uma lista de construções IDs

O exemplo de declaração de política a seguir permite que um usuário obtenha uma lista de todas IDs as versões da mesma conta:

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

```
    "Resource": "*"
  }
]
}
```

Permitir que um usuário obtenha uma lista de frotas

O seguinte exemplo de declaração de políticas permite que um usuário obtenha uma lista de frotas na região us-east-2, para a conta 123456789012:

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

Permitir que um usuário obtenha uma lista de grupos de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário obtenha uma lista de grupos de relatórios na região us-east-2, para a conta 123456789012:

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

Permitir que um usuário obtenha uma lista de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário obtenha uma lista de relatórios na região us-east-2, para a conta 123456789012:

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

Permitir que um usuário obtenha uma lista de relatórios para um grupo de relatórios

O seguinte exemplo de declaração de políticas permite que um usuário obtenha uma lista de relatórios para um grupo de relatórios na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportsForReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que um usuário obtenha uma lista de casos de teste para um relatório

O seguinte exemplo de declaração de políticas permite que um usuário obtenha uma lista de casos de teste para um relatório na região us-east-2, para a conta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DescribeTestCases",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```


Permitir que um usuário comece a executar compilações

O seguinte exemplo de declaração de políticas permite que um usuário execute compilações na região us-east-2, para a conta 123456789012 para um projeto de compilação que comece com o nome my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que um usuário tente parar compilações

O seguinte exemplo de declaração de política permite que um usuário tente parar a execução de builds somente na região us-east-2, para a conta 123456789012, para qualquer projeto de build que comece com o nome my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que um usuário tente excluir compilações

Ao seguinte exemplo de declaração de política permite que um usuário tente excluir compilações somente na região us-east-2 da conta 123456789012 de qualquer projeto de compilação que comece com o nome my:

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": "codebuild:BatchDeleteBuilds",
        "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
      }
    ]
  }
}

```

Permitir que um usuário obtenha informações sobre imagens do Docker que são gerenciadas pelo CodeBuild

O seguinte exemplo de declaração de políticas permite que um usuário obtenha informações sobre todas as imagens Docker que são gerenciadas pelo CodeBuild:

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

Permitir que um usuário adicione uma política de permissão para um perfil de serviço de frota

O exemplo de declaração de política de recursos a seguir permite que um usuário adicione uma política de permissão da VPC para um perfil de serviço de frota:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVpcCreateNI",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:region:account-id:subnet/subnet-id-1",

```

```

    "arn:aws:ec2:region:account-id:security-group/security-group-id-1",
    "arn:aws:ec2:region:account-id:network-interface/*"
  ],
},
{
  "Sid": "CodeBuildFleetVpcPermission",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcs",
    "ec2:ModifyNetworkInterfaceAttribute",
    "ec2>DeleteNetworkInterface"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeBuildFleetVpcNIPermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:region:account-id:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:Subnet": [
        "arn:aws:ec2:region:account-id:subnet/subnet-id-1"
      ]
    }
  }
}
]
}

```

O exemplo de declaração de política de recursos a seguir permite que um usuário adicione uma política de permissão personalizada da Amazon Managed Image (AMI) para um perfil de serviço de frota:

```

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

```

```
{
  "Effect": "Allow",
  "Action": "ec2:DescribeImages",
  "Resource": "*"
}
```

O exemplo de declaração de política de confiança a seguir permite a um usuário adicionar uma política de permissão para um perfil de serviço de frota:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVPCTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

Permitir CodeBuild acesso aos AWS serviços necessários para criar uma interface de rede VPC

O exemplo de declaração de política a seguir concede AWS CodeBuild permissão para criar uma interface de rede em uma VPC com duas sub-redes:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",

```

```

        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:region:account-id:network-interface/*",
    "Condition": {
        "StringEquals": {
            "ec2:AuthorizedService": "codebuild.amazonaws.com"
        },
        "ArnEquals": {
            "ec2:Subnet": [
                "arn:aws:ec2:region:account-id:subnet/subnet-id-1",
                "arn:aws:ec2:region:account-id:subnet/subnet-id-2"
            ]
        }
    }
}
]
}

```

Use uma declaração de negação para AWS CodeBuild evitar a desconexão dos provedores de origem

O exemplo de declaração de política a seguir usa uma instrução de negação para impedir que o AWS CodeBuild se desconecte dos provedores de origem. Ele usa `codebuild:DeleteAuthToken`, que é o inverso de `codebuild:PersistAuthToken` e `codebuild:ImportSourceCredentials`, para se conectar com os provedores de origem. Para obter mais informações, consulte [Permissões necessárias para que o AWS CodeBuild console se conecte aos provedores de origem](#).

```

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

```

```
{
  "Effect": "Deny",
  "Action": "codebuild:DeleteOAuthToken",
  "Resource": "*"
}
```

AWS CodeBuild referência de permissões

Você pode usar chaves AWS de condição abrangentes em suas AWS CodeBuild políticas para expressar condições. Para obter uma lista, consulte [Available Keys](#) no Guia do usuário do IAM.

Você especifica as ações no campo `Action` das políticas. Para especificar uma ação, use o prefixo `codebuild:` seguido pelo nome da operação API (por exemplo, `codebuild:CreateProject` e `codebuild:StartBuild`). Para especificar várias ações em uma única declaração, separe-as com vírgulas (por exemplo, `"Action": [ "codebuild:CreateProject", "codebuild:StartBuild" ]`).

Usando caracteres curinga

Você especifica um ARN, com ou sem um caractere curinga (*), como o valor do recurso no campo `Resource` das políticas. Você pode usar um curinga para especificar várias ações ou recursos. Por exemplo, `codebuild:*` especifica todas as CodeBuild ações e `codebuild:Batch*` especifica todas as CodeBuild ações que começam com a palavra. `Batch` O seguinte exemplo concede acesso a qualquer projeto de build com nomes que comecem com `my`:

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

CodeBuild Operações de API e permissões necessárias para ações

BatchDeleteBuilds

Ação: `codebuild:BatchDeleteBuilds`

Necessários para excluir compilações.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetBuilds

Ação: `codebuild:BatchGetBuilds`

Exigido para obter informações sobre builds.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetProjects

Ação: `codebuild:BatchGetProjects`

Exigido para obter informações sobre projetos de build.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetReportGroups

Ação: `codebuild:BatchGetReportGroups`

Necessário para obter informações sobre grupos de relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchGetReports

Ação: `codebuild:BatchGetReports`

Necessário para obter informações sobre relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchPutTestCases¹

Ação: `codebuild:BatchPutTestCases`

Necessário para criar ou atualizar um relatório de teste.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateProject

Ações: `codebuild>CreateProject`, `iam:PassRole`

Exigido para criar projetos de build.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`

- `arn:aws:iam::account-ID:role/role-name`

CreateReport¹

Ação: `codebuild:CreateReport`

Necessário para criar um relatório de teste.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateReportGroup

Ação: `codebuild:CreateReportGroup`

Necessário para criar um grupo de relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateWebhook

Ação: `codebuild:CreateWebhook`

Necessário para criar um webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteProject

Ação: `codebuild>DeleteProject`

Necessário para excluir um CodeBuild projeto.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteReport

Ação: `codebuild>DeleteReport`

Necessário para excluir um relatório.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteReportGroup

Ação: `codebuild>DeleteReportGroup`

Necessário para excluir um grupo de relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteSourceCredentials

Ação: `codebuild>DeleteSourceCredentials`

Necessário para excluir um conjunto de `SourceCredentialsInfo` objetos que contêm informações sobre credenciais para um repositório GitHub, GitHub Enterprise Server ou Bitbucket.

Recurso: *

DeleteWebhook

Ação: `codebuild>DeleteWebhook`

Necessário para criar um webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DescribeTestCases

Ação: `codebuild>DescribeTestCases`

Necessário para retornar uma lista paginada de casos de teste.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

ImportSourceCredentials

Ação: `codebuild>ImportSourceCredentials`

Necessário para importar um conjunto de `SourceCredentialsInfo` objetos que contenham informações sobre credenciais para um repositório GitHub, GitHub Enterprise Server ou Bitbucket.

Recurso: *

InvalidateProjectCache

Ação: `codebuild>InvalidateProjectCache`

Necessário para redefinir o cache para um projeto.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuildBatches

Ação: `codebuild:ListBuildBatches`

Necessário para obter uma lista do lote de compilação IDs.

Recurso: *

ListBuildBatchesForProject

Ação: `codebuild:ListBuildBatchesForProject`

Necessário para obter uma lista de lotes de compilação IDs para um projeto específico.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuilds

Ação: `codebuild:ListBuilds`

Necessário para obter uma lista de construções IDs.

Recurso: *

ListBuildsForProject

Ação: `codebuild:ListBuildsForProject`

Necessário para obter uma lista de IDs compilações para um projeto de construção.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListCuratedEnvironmentImages

Ação: `codebuild:ListCuratedEnvironmentImages`

Exigido para obter informações sobre todas as imagens Docker que são gerenciadas pelo AWS CodeBuild.

Recurso: * (necessário, mas não se refere a um recurso da AWS endereçável)

ListProjects

Ação: `codebuild:ListProjects`

Exigido para obter uma lista de nomes de projeto de build.

Recurso: *

ListReportGroups

Ação: `codebuild:ListReportGroups`

Necessário para obter uma lista de grupos de relatórios.

Recurso: *

ListReports

Ação: `codebuild:ListReports`

Necessário para obter uma lista de relatórios.

Recurso: *

ListReportsForReportGroup

Ação: `codebuild:ListReportsForReportGroup`

Necessário para obter uma lista de relatórios para um grupo de relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

RetryBuild

Ação: `codebuild:RetryBuild`

Necessário para repetir as compilações.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StartBuild

Ação: `codebuild:StartBuild`

Exigido para começar a executar builds.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StopBuild

Ação: `codebuild:StopBuild`

Exigido para tentar parar de executar builds.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

UpdateProject

Ações: `codebuild:UpdateProject`, `iam:PassRole`

Exigido para alterar informações sobre builds.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateProjectVisibility

Ações: `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Necessário para alterar a visibilidade pública das compilações de um projeto.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateReport¹

Ação: `codebuild:UpdateReport`

Necessário para criar ou atualizar um relatório de teste.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateReportGroup

Ação: `codebuild:UpdateReportGroup`

Necessário para atualizar um grupo de relatórios.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateWebhook

Ação: `codebuild:UpdateWebhook`

Necessário para atualizar um webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

¹ Usado apenas para permissão. Não há API para esta ação.

Usar tags para controlar o acesso aos recursos do AWS CodeBuild

As condições nas declarações de política do IAM fazem parte da sintaxe que você pode usar para especificar permissões para ações baseadas em CodeBuild projetos. É possível criar uma política que permita ou negue ações em projetos com base nas tags associadas a esses projetos e aplicar essas políticas aos grupos do IAM que você configura para gerenciar usuários. Para obter informações sobre como aplicar tags a um projeto usando o console ou AWS CLI, consulte [Crie um projeto de construção em AWS CodeBuild](#). Para obter informações sobre como aplicar tags usando o CodeBuild SDK, consulte [CreateProject Tags](#) na Referência da CodeBuild API. Para obter informações sobre o uso de tags para controlar o acesso aos AWS recursos, consulte [Como controlar o acesso aos AWS recursos usando tags](#) de recursos no Guia do usuário do IAM.

Important

Ao usar o recurso de capacidade reservada, os dados armazenados em cache nas instâncias da frota, incluindo arquivos de origem, camadas do Docker e diretórios em cache especificados no buildspec, podem ser acessados por outros projetos na mesma conta. Isso ocorre por design e permite que projetos dentro da mesma conta compartilhem instâncias de frota.

Exemplo Exemplo 1: limitar as ações CodeBuild do projeto com base nas tags de recursos

O exemplo a seguir nega todas as ações `BatchGetProjects` em projetos marcados com a chave `Environment` com o valor de chave de `Production`. O administrador de um usuário deve anexar essa política do IAM, além de uma política gerenciada pelo usuário, a usuários não autorizados. A chave de condição `aws:ResourceTag` é usada para controlar o acesso a recursos com base em suas tags.

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

```

    "Action": [
      "codebuild:BatchGetProjects"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/Environment": "Production"
      }
    }
  }
]
}

```

Example Exemplo 2: limitar as ações CodeBuild do projeto com base nas tags de solicitação

A política a seguir negará a permissão dos usuários para a ação `CreateProject` se a solicitação tiver uma tag com a chave `Environment` e o valor de chave `Production`. Além disso, a política impede que esses usuários não autorizados modifiquem projetos usando a chave de condição `aws:TagKeys` para não permitir `UpdateProject` se a solicitação tiver uma tag com a chave `Environment`. Um administrador deve anexar essa política do IAM, além da política gerenciada pelo usuário, a usuários que não estão autorizados a executar essas ações. A chave de condição `aws:RequestTag` é usada para controlar quais tags podem ser transmitidas em uma solicitação do IAM.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:CreateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/Environment": "Production"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [

```

```

    "codebuild:UpdateProject"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:TagKeys": ["Environment"]
    }
  }
}
]
}

```

Example Exemplo 3: Negar ou permitir ações em grupos de relatórios com base em tags de recursos

Você pode criar uma política que permita ou negue ações em CodeBuild recursos (projetos e grupos de relatórios) com base nas AWS tags associadas a esses recursos e, em seguida, aplicar essas políticas aos grupos do IAM que você configura para gerenciar usuários. Por exemplo, você pode criar uma política que negue todas as CodeBuild ações em qualquer grupo de relatórios com a chave de AWS tag Status e o valor da chave de eSecret, 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 grupos de relatórios 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 CodeBuild ações em grupos de relatórios marcados com a chave Status e o valor-chave deSecret:

```

{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild>DeleteReportGroup",
        "codebuild>ListReportGroups",
        "codebuild>ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
      ]
      "Resource" : "*",
    }
  ]
}

```

```

    "Condition" : {
      "StringEquals" : "aws:ResourceTag/Status": "Secret"
    }
  }
]
}

```

Exemplo 4: limitar CodeBuild ações AWSCode BuildDeveloperAccess com base em tags de recursos

Você pode criar políticas que permitam CodeBuild ações em todos os grupos de relatórios e projetos que não estejam marcados com tags específicas. Por exemplo, a política a seguir permite o equivalente de permissões [AWSCodeBuildDeveloperAccess](#) para todos os grupos de relatórios e projetos, exceto os marcados com as tags especificadas:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceTag/Status": "Secret",

```



```
    "aws:ResourceTag/Team": "Saanvi"  
  }  
}  
}  
]  
}
```

Visualizar recursos no console

O AWS CodeBuild console exige a `ListRepositories` permissão para exibir uma lista de repositórios para sua AWS conta na AWS região 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 em sua AWS conta na AWS região 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, alguns recursos não serão retornados se houver um `Deny` explícito para visualizar esses recursos.

Validação de conformidade para AWS CodeBuild

Audidores terceirizados avaliam a segurança e a conformidade AWS CodeBuild como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte [AWS serviços no escopo por programa de conformidade](#). Para obter informações gerais, consulte [Programas de conformidade da AWS](#).

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Como [baixar relatórios no AWS Artifact](#).

Sua responsabilidade de conformidade ao usar CodeBuild é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Se seu uso CodeBuild estiver sujeito à conformidade com padrões como HIPAA, PCI ou FedRAMP, fornece recursos para ajudar a: AWS

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados em segurança e conformidade em AWS.
- Documento técnico [sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper](#) descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- [AWS recursos de conformidade](#) — essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Config](#) — Esse AWS serviço avalia se suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Monitore seu uso das AWS CodeBuild melhores práticas de segurança no que se refere às melhores práticas de segurança usando [AWS Security Hub](#). O Security Hub usa controles de segurança para avaliar configurações de recursos e padrões de segurança que ajudam você a cumprir vários frameworks de conformidade. Para obter mais informações sobre como usar o Security Hub para avaliar CodeBuild recursos, consulte [AWS CodeBuild os controles](#) no Guia AWS Security Hub do Usuário.

Resiliência em AWS CodeBuild

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de

disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [infraestrutura AWS global](#).

Segurança da infraestrutura em AWS CodeBuild

Como serviço gerenciado, AWS CodeBuild é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar CodeBuild pela 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.

Além disso, 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 é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Acesse seu provedor de origem em CodeBuild

Para GitHub nosso GitHub Enterprise Server, você usa um token de acesso pessoal, um segredo do Secrets Manager, uma conexão ou um OAuth aplicativo para acessar o provedor de origem. Para o Bitbucket, você usa um token de acesso, uma senha de aplicativo, um segredo do Secrets Manager, uma conexão ou um OAuth aplicativo para acessar o provedor de origem.

Tópicos

- [Criar e armazenar um token em um segredo do Secrets Manager](#)
- [GitHub e acesso ao GitHub Enterprise Server em CodeBuild](#)

- [Acesso ao Bitbucket em CodeBuild](#)
- [GitLab acesso em CodeBuild](#)

Criar e armazenar um token em um segredo do Secrets Manager

Se você optar por armazenar seu token de acesso usando o Secrets Manager, poderá usar uma conexão secreta existente ou criar um segredo. Para criar um segredo, faça o seguinte:

AWS Management Console

Para criar um segredo do Secrets Manager no AWS Management Console

1. Em Provedor de origem, escolha Bitbucket ou GitHub Enterprise. GitHub
2. Em Credencial, siga um destes procedimentos:
 - Escolha Credencial de origem padrão para usar a credencial de origem padrão da conta e aplicá-la a todos os projetos.
 - a. Se você ainda não tiver feito login no provedor de origem, escolha Gerenciar credencial de origem padrão.
 - b. Em Tipo de credencial, escolha um tipo de credencial diferente de. CodeConnections
 - c. Em Serviço, escolha Secrets Manager e, para Segredos, escolha Novo segredo.
 - d. Em Nome do segredo, insira o nome do segredo.
 - e. Em Descrição do segredo - opcional, insira uma descrição para o segredo.
 - f. Dependendo do provedor de origem escolhido, insira o token ou o nome de usuário e a senha da aplicação e escolha Salvar.
 - Escolha Credencial de origem personalizada para usar uma credencial de origem personalizada para substituir as configurações padrão da conta.
 - a. Em Tipo de credencial, escolha um tipo de credencial diferente de. CodeConnections
 - b. Em Conexão, escolha Criar um segredo.
 - c. Em Nome do segredo, insira o nome do segredo.
 - d. Em Descrição do segredo - opcional, insira uma descrição para o segredo.

- e. Dependendo do provedor de origem escolhido, insira o token ou o nome de usuário e a senha da aplicação e escolha Criar.

AWS CLI

Para criar um segredo do Secrets Manager no AWS CLI

- Abra um terminal (Linux, macOS ou Unix) ou um prompt de comando (Windows). Use o AWS CLI para executar o create-secret comando Secrets Manager.

```
aws secretsmanager create-secret --region <aws-region> \
    --name '<secret-name>' \
    --description '<secret-description>' \
    --secret-string '{
        "ServerType": "<server-type>",
        "AuthType": "<auth-type>",
        "Token": "<token>"
    }' \
    --tags Key=codebuild:source,Value='' \
        Key=codebuild:source:type,Value=<type> \
        Key=codebuild:source:provider,Value=<provider>
```

Os segredos do Secrets Manager CodeBuild aceitos devem estar na mesma conta e AWS região do CodeBuild projeto e devem estar no seguinte formato JSON:

```
{
    "ServerType": ServerType,
    "AuthType": AuthType,
    "Token": string,
    "Username": string // Optional and is only used for Bitbucket app
    password
}
```

Campo	Valores válidos	Descrição
ServerType	GITHUB GITHUB_ENTERPRISE BITBUCKET	O provedor de origem de terceiros do segredo do Secrets Manager.
AuthType	PERSONAL_ACCESS_TOKEN BASIC_AUTH	O tipo de token de acesso usado pelas credenciais. Pois GitHub, somente PERSONAL_ACCESS_TOKEN é válido. BASIC_AUTH só é válido para a senha da aplicação do Bitbucket.
Token	<i>string</i>	Para GitHub nossa GitHub empresa, esse é o token de acesso pessoal. Para o Bitbucket, esse é o token de acesso ou a senha da aplicação do Bitbucket.
Nome de usuário	<i>string</i>	O nome de usuário do Bitbucket quando AuthType é BASIC_AUTH. Esse parâmetro não é válido para outros tipos de provedores de origem.

Além disso, CodeBuild usa as seguintes tags de recursos no segredo para garantir que os segredos sejam facilmente selecionáveis ao criar ou editar projetos.

Chave de tag	Valor da tag	Descrição
codebuild:source:provider	github	Diz para CodeBuild qual provedor esse segredo se destina.
	github_enterprise	
	bitbucket	
codebuild:source:type	personal_access_token	Informa CodeBuild o tipo de token de acesso nesse segredo.
	basic_auth	

GitHub e acesso ao GitHub Enterprise Server em CodeBuild

Pois GitHub, você pode usar um token de acesso pessoal, um OAuth aplicativo, um segredo do Secrets Manager ou uma conexão de GitHub aplicativo para acessar o provedor de origem. Para o GitHub Enterprise Server, você pode usar um token de acesso pessoal, um segredo do Secrets Manager ou uma conexão de GitHub aplicativo para acessar o provedor de origem.

Tópicos

- [GitHub Conexões de aplicativos para GitHub um GitHub Enterprise Server](#)
- [GitHub e token de acesso do GitHub Enterprise Server](#)
- [GitHub OAuth Aplicativo](#)

GitHub Conexões de aplicativos para GitHub um GitHub Enterprise Server

Você pode usar o GitHub aplicativo para se conectar com CodeBuild. GitHub As conexões de aplicativos são suportadas por meio de [Conexões de código da AWS](#).

O acesso do provedor de origem permite que você acione uma compilação ao assinar [CreateWebhook](#), [GitHub eventos de webhook](#) usar ou usar [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#) em CodeBuild.

 Note

CodeConnections está disponível em menos regiões do que CodeBuild. Você pode usar conexões entre regiões em CodeBuild. Conexões criadas em regiões de aceitação não podem ser usadas em outras regiões. Para obter mais informações, consulte [Endpoints e cotas do Conexões de código da AWS](#).

Tópicos

- [Etapa 1: criar uma conexão com o GitHub aplicativo \(console\)](#)
- [Etapa 2: conceder acesso à função IAM do CodeBuild projeto para usar a conexão](#)
- [Etapa 3: Configurar CodeBuild para usar a nova conexão](#)
- [Solução de problemas com o GitHub aplicativo](#)

Etapa 1: criar uma conexão com o GitHub aplicativo (console)

Use essas etapas para usar o CodeBuild console para adicionar uma conexão ao seu projeto no GitHub.

Para criar uma conexão com GitHub

- Siga as instruções no Guia do usuário das ferramentas do desenvolvedor para [criar uma conexão com GitHub](#).

 Note

Em vez de criar ou usar uma conexão existente na sua conta, você pode usar uma conexão compartilhada de outra AWS conta. Para obter mais informações, consulte [Compartilhar conexões com AWS contas](#).

Etapa 2: conceder acesso à função IAM do CodeBuild projeto para usar a conexão

Você pode conceder acesso à função IAM do CodeBuild projeto para usar os GitHub tokens fornecidos pela sua conexão.

Para conceder acesso à função IAM do CodeBuild projeto

1. Crie uma função do IAM para seu CodeBuild projeto seguindo as instruções [CodeBuild Permitir interagir com outros AWS serviços](#) para seu CodeBuild projeto.
2. Ao seguir as instruções, adicione a seguinte política do IAM ao papel CodeBuild do seu projeto para conceder acesso à conexão.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codeconnections:GetConnectionToken",
        "codeconnections:GetConnection"
      ],
      "Resource": [
        "<connection-arn>"
      ]
    }
  ]
}
```

Etapa 3: Configurar CodeBuild para usar a nova conexão

Você pode configurar uma conexão como uma credencial no nível da conta e usá-la em um projeto.

AWS Management Console

Para configurar uma conexão como uma credencial de nível de conta no AWS Management Console

1. Em Source provider, escolha GitHub.
2. Em Credencial, siga um destes procedimentos:
 - Escolha Credencial de origem padrão para usar a credencial de origem padrão da conta e aplicá-la a todos os projetos.
 - a. Se você não estiver conectado GitHub, escolha Gerenciar credencial de origem padrão.

- b. Em Tipo de credencial, escolha GitHub Aplicativo.
- c. Em Conexão, escolha usar uma conexão existente ou crie uma.
- Escolha Credencial de origem personalizada para usar uma credencial de origem personalizada para substituir as configurações padrão da conta.
 - a. Em Tipo de credencial, escolha GitHub Aplicativo.
 - b. Em Conexão, escolha usar uma conexão existente ou crie uma.

AWS CLI

Para configurar uma conexão como uma credencial de nível de conta no AWS CLI

- Abra um terminal (Linux, macOS ou Unix) ou um prompt de comando (Windows). Use o AWS CLI para executar o `import-source-credentials` comando, especificando o `--auth-type` `--server-type`, e `--token` para sua conexão.

Use o seguinte comando:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type GITHUB --token <connection-arn>
```

Você também pode configurar vários tokens para seus CodeBuild projetos. Para obter mais informações, consulte [Configurar vários tokens como credenciais no nível de origem](#).

Solução de problemas com o GitHub aplicativo

As informações a seguir podem ajudá-lo a solucionar problemas comuns com o GitHub aplicativo.

Tópicos

- [Instale o AWS conector para GitHub aplicativo em uma região indesejada](#)
- [A conexão do GitHub aplicativo não tem acesso aos repositórios](#)
- [A função do IAM do AWS serviço não tem as permissões necessárias do IAM.](#)

Instale o AWS conector para GitHub aplicativo em uma região indesejada

Problema: você instalou o AWS Connector GitHub do GitHub Marketplace, mas a conexão foi criada em uma região indesejada. Se você tentar reconfigurar o aplicativo no GitHub site, ele não funcionará porque o aplicativo já está instalado na sua GitHub conta.

Possível causa: o aplicativo já está instalado em sua GitHub conta, então você só pode reconfigurar as permissões do aplicativo.

Solução recomendada: você pode criar uma conexão com o ID de instalação na região desejada.

1. Abra o CodeConnections console em <https://console.aws.amazon.com/codesuite/configurações/conexões> e navegue até a região desejada usando o seletor de região na barra de navegação do AWS console.
2. Siga as instruções no Guia do usuário das ferramentas do desenvolvedor para [criar uma conexão com GitHub](#).

Note

Como você já instalou o AWS Connector for GitHub app, você pode escolhê-lo em vez de instalar um novo aplicativo.

A conexão do GitHub aplicativo não tem acesso aos repositórios

Problema: um AWS serviço usando a conexão, como CodeBuild ou CodePipeline, relata que não tem acesso ao repositório ou que o repositório não existe. Algumas possíveis mensagens de erro incluem:

- Authentication required for primary source.
- Unable to create webhook at this time. Please try again later.
- Failed to create webhook. GitHub API limit reached. Please try again later.

Possível causa: você pode estar usando o GitHub aplicativo e não ter concedido o escopo da permissão do webhook.

Solução recomendada: para conceder o escopo de permissão necessário, siga as instruções em [Navegar até o GitHub aplicativo que você deseja revisar ou modificar](#) para configurar o

aplicativo instalado. Na seção de permissões, você verá que a aplicação não tem permissão para webhooks e há uma opção para você revisar as permissões recém-solicitadas. Revise e aceite as novas permissões. Para obter mais informações, consulte [Aprovação de permissões atualizadas para um GitHub aplicativo](#).

Possível causa: a conexão estava funcionando conforme o esperado, mas repentinamente não tem acesso aos repositórios.

Solução possível: comece revisando suas [autorizações](#) e [instalações](#) e, em seguida, verifique se o GitHub aplicativo está autorizado e instalado. Se a instalação do GitHub aplicativo estiver suspensa, você precisará cancelá-la. Se o GitHub aplicativo não estiver autorizado para uma conexão [UAT \(User Access Token\)](#) ou não estiver instalado para uma conexão [IAT \(Installation Access Token\)](#), a conexão existente não poderá mais ser usada e você precisará criar uma nova conexão. Observe que a reinstalação do GitHub aplicativo não reativará a conexão anterior associada à instalação antiga.

Solução possível: se a conexão for uma conexão UAT, certifique-se de que a conexão não esteja sendo usada simultaneamente, como sendo usada em várias execuções CodeBuild simultâneas de compilação. Isso ocorre porque invalida GitHub imediatamente um UAT emitido anteriormente se um token expirado for atualizado pela conexão. Se você precisar usar a conexão UAT para várias CodeBuild compilações simultâneas, poderá criar várias conexões e usar cada conexão de forma independente.

Solução possível: se a conexão UAT não tiver sido usada nos últimos 6 meses, a conexão será invalidada pelo. GitHub Para corrigir isso, crie uma conexão.

Possível causa: você pode estar usando uma conexão UAT sem instalar a aplicação.

Solução recomendada: embora a criação de uma conexão UAT não exija a associação da conexão a uma instalação do GitHub aplicativo, é necessária uma instalação para que o repositório esteja acessível. Siga as instruções para [revisar as instalações](#) e garantir que o GitHub aplicativo esteja instalado. Se não estiver instalado, navegue até a [página do GitHub aplicativo](#) para instalar o aplicativo. Para obter mais informações sobre o acesso do UAT, consulte [Sobre os tokens de acesso do usuário](#).

A função do IAM do AWS serviço não tem as permissões necessárias do IAM.

Problema: você vê uma das seguintes mensagens de erro:

- Access denied to connection *<connection-arn>*

- Failed to get access token from `<connection-arn>`

Solução recomendada: Normalmente, você usa uma conexão com um AWS serviço, como CodePipeline ou CodeBuild. Quando você atribui ao AWS serviço uma função do IAM, o AWS serviço pode usar a permissão da função para agir em seu nome. Verifique se o perfil do IAM tem as permissões necessárias. Para obter mais informações sobre a permissão necessária do IAM, consulte [Conceder acesso ao papel IAM do CodeBuild projeto para usar a conexão](#) e o [gerenciamento de identidade e acesso para AWS CodeStar notificações e CodeConnections](#) no Guia do usuário do console Developer Tools.

GitHub e token de acesso do GitHub Enterprise Server

Pré-requisitos do token de acesso

Antes de começar, você deve adicionar os escopos de permissão adequados ao seu token de GitHub acesso.

Pois GitHub, seu token de acesso pessoal deve ter os seguintes escopos.

- `repo`: concede controle total de repositórios privados.
- `repo:status`: concede read/write acesso aos status de confirmação do repositório público e privado.
- `admin:repo_hook`: concede controle total de ganchos do repositório. Esse escopo não será obrigatório se o token tiver o escopo `repo`.
- `admin:org_hook`: concede controle total dos hooks da organização. Esse escopo só é necessário se você estiver usando o recurso de webhook da organização.

Para obter mais informações, consulte [Entendendo os escopos dos OAuth aplicativos](#) no GitHub site.

Se você estiver usando tokens de acesso pessoal refinados, dependendo do caso de uso, o token de acesso pessoal pode precisar das seguintes permissões:

- `Conteúdo: somente leitura`: concede acesso a repositórios privados. Essa permissão é necessária se você estiver usando repositórios privados como fonte.
- `Status de confirmação: leitura e gravação`: concede permissão para criar status de confirmação. Essa permissão é necessária se o projeto tiver um webhook configurado ou se você tiver o recurso de status de criação de relatório habilitado.
- `Webhooks: leitura e gravação`: concede permissão para gerenciar webhooks. Essa permissão é necessária se o seu projeto tiver um webhook configurado.

- Solicitações pull: somente leitura: concede permissão para acessar solicitações pull. Essa permissão é necessária se o webhook tiver um filtro FILE_PATH em eventos de solicitação pull.
- Administração: leitura e gravação: essa permissão é necessária se você estiver usando o recurso de execução de GitHub ações auto-hospedado com CodeBuild. Para obter mais detalhes, consulte [Criar um token de registro para um repositório](#) e [Tutorial: Configurar um CodeBuild executor de GitHub ações hospedado](#).

 Note

Se você quiser acessar os repositórios da organização, especifique a organização como proprietária do recurso do token de acesso.

Para obter mais informações, consulte [Permissões necessárias para tokens de acesso pessoal refinados no site](#). GitHub

Conecte-se GitHub com um token de acesso (console)

Para usar o console para conectar seu projeto ao GitHub uso de um token de acesso, faça o seguinte ao criar um projeto. Para mais informações, consulte [Criar um projeto de compilação \(console\)](#).

1. Em Source provider, escolha GitHub.
2. Em Credencial, siga um destes procedimentos:
 - Escolha usar as credenciais da conta para aplicar a credencial de origem padrão da sua conta a todos os projetos.
 - a. Se você não estiver conectado GitHub, escolha Gerenciar credencial da conta.
 - b. Em Tipo de credencial, escolha Token de acesso pessoal.
 - Se você optar por usar credenciais de nível de conta para o Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:
 - a. Se você optar por usar o Secrets Manager, poderá optar por usar uma conexão secreta existente ou criar um novo segredo e, em seguida, escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).

- b. Se você optar por usar CodeBuild, insira seu token de acesso GitHub pessoal e escolha Salvar.
- Selecione Usar credenciais de substituição somente para este projeto para usar uma credencial de origem personalizada para substituir as configurações de credencial da sua conta.
 - a. Na lista de credenciais preenchida, escolha uma das opções em Token de acesso pessoal.
 - b. Você também pode criar um novo token de acesso pessoal selecionando criar uma nova conexão de token de acesso pessoal na descrição.

Conecte-se GitHub com um token de acesso (CLI)

Siga estas etapas para usar o AWS CLI para conectar seu projeto ao GitHub uso de um token de acesso. Para obter informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

1. Execute o comando import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Os dados formatados JSON aparecem na saída. Copie os dados para um arquivo (por exemplo, *import-source-credentials.json*) em um local no computador local ou na instância em que o AWS CLI está instalado. Modifique os dados copiados da maneira a seguir e salve os resultados.

```
{
  "serverType": "server-type",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Substitua o seguinte:

- ***server-type***: Valor obrigatório. O provedor de origem usado para essa credencial. Os valores válidos são GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB e GITLAB_SELF_MANAGED.
 - ***auth-type***: Valor obrigatório. O tipo de autenticação usado para se conectar a um repositório. Os valores válidos são OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS e SECRETS_MANAGER. Pois GitHub, somente PERSONAL_ACCESS_TOKEN é permitido. BASIC_AUTH só é permitido com a senha da aplicação do Bitbucket.
 - ***should-overwrite***: Valor opcional. Defina como false para impedir a substituição das credenciais de origem do repositório. Defina como true para substituir as credenciais de origem do repositório. O valor padrão é true.
 - ***token***: Valor obrigatório. Para GitHub nosso GitHub Enterprise Server, esse é o token de acesso pessoal. Para o Bitbucket, esse é o token de acesso pessoal ou senha da aplicação. Para o tipo de autenticação CODECONNECTIONS, esse é o ARN da conexão. Para o tipo de autenticação SECRETS_MANAGER, esse é o ARN secreto.
 - ***username***: Valor opcional. Esse parâmetro é ignorado para provedores GitHub de origem do GitHub Enterprise Server.
2. Para conectar sua conta com um token de acesso, alterne para o diretório que contém o arquivo `import-source-credentials.json` que você salvou na etapa 1 e execute o comando `import-source-credentials` novamente.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Dados no formato JSON são exibidos na saída com um Nome de recurso da Amazon (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

 Note

Se você executar o comando `import-source-credentials` com o mesmo tipo de servidor e tipo de autenticação uma segunda vez, o token de acesso armazenado será atualizado.

Depois que sua conta estiver conectada a um token de acesso, você poderá usá-lo `create-project` para criar seu CodeBuild projeto. Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).

3. Para visualizar os tokens de acesso conectados, execute o comando `list-source-credentials`.

```
aws codebuild list-source-credentials
```

O objeto no formato JSON `sourceCredentialsInfos` é exibido na saída:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "server-type",
      "arn": "arn"
    }
  ]
}
```

`sourceCredentialsObject` contém uma lista de informações de credenciais de origem conectadas:

- `authType` é o tipo de autenticação usada pelas credenciais. Pode ser `OAUTH`, `BASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS` ou `SECRETS_MANAGER`.
 - `serverType` é o tipo de provedor de origem. Pode ser `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB` ou `GITLAB_SELF_MANAGED`.
 - `arn` é o ARN do token.
4. Para se desconectar de um provedor de origem e remover seus tokens de acesso, execute o comando `delete-source-credentials` com seu ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Dados no formato JSON são retornados com um ARN das credenciais excluídas.

```
{
```

```
"arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

GitHub OAuth Aplicativo

Conecte-se GitHub usando OAuth (console)

Para usar o console para conectar seu projeto ao GitHub uso de um OAuth aplicativo, faça o seguinte ao criar um projeto. Para mais informações, consulte [Criar um projeto de compilação \(console\)](#).

1. Em Source provider, escolha GitHub.
2. Em Credencial, siga um destes procedimentos:
 - Escolha usar as credenciais da conta para aplicar a credencial de origem padrão da sua conta a todos os projetos.
 - a. Se você não estiver conectado GitHub, escolha Gerenciar credencial da conta.
 - b. Para Tipo de credencial, escolha OAuth aplicativo.
 - Se você optar por usar credenciais de nível de conta para o Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:
 - a. Se você optar por usar o Secrets Manager, poderá optar por usar uma conexão secreta existente ou criar um novo segredo e, em seguida, escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).
 - b. Se você optar por usar CodeBuilde, em seguida, escolher Salvar.
 - Selecione Usar credenciais de substituição somente para este projeto para usar uma credencial de origem personalizada para substituir as configurações de credencial da sua conta.
 - a. Na lista de credenciais preenchida, escolha uma das opções em OAuth app.
 - b. Você também pode criar um novo token de OAuth aplicativo selecionando criar uma nova conexão de token de aplicativo Oauth na descrição.

Para revisar seus OAuth aplicativos autorizados, navegue até [Aplicativos ativados](#) GitHub e verifique se um aplicativo chamado de AWS CodeBuild (*region*) propriedade da [aws-codesuite](#) está listado.

Acesso ao Bitbucket em CodeBuild

Para o Bitbucket, você usa um token de acesso, uma senha de aplicativo, um OAuth aplicativo ou uma conexão do Bitbucket para acessar o provedor de origem.

Tópicos

- [Conexões da aplicação do Bitbucket](#)
- [Senha da aplicação ou token de acesso do Bitbucket](#)
- [Aplicativo Bitbucket OAuth](#)

Conexões da aplicação do Bitbucket

Você pode usar o Bitbucket para se conectar com CodeBuild. As conexões da aplicação do Bitbucket têm suporte por meio do [Conexões de código da AWS](#).

Note

CodeConnections está disponível em menos regiões do que CodeBuild. Você pode usar conexões entre regiões em CodeBuild. Conexões criadas em regiões de aceitação não podem ser usadas em outras regiões. Para obter mais informações, consulte [Endpoints e cotas do Conexões de código da AWS](#).

Tópicos

- [Etapa 1: criar uma conexão com o Bitbucket \(console\)](#)
- [Etapa 2: conceder acesso à função IAM do CodeBuild projeto para usar a conexão](#)
- [Etapa 3: Configurar CodeBuild para usar a nova conexão](#)

Etapa 1: criar uma conexão com o Bitbucket (console)

Use essas etapas para usar o CodeBuild console para adicionar uma conexão ao seu projeto no Bitbucket.

Para criar uma conexão com o Bitbucket

- Siga as instruções no Guia do usuário do Developer Tools para [Criar uma conexão com o Bitbucket](#).

 Note

Em vez de criar ou usar uma conexão existente na sua conta, você pode usar uma conexão compartilhada de outra AWS conta. Para obter mais informações, consulte [Compartilhar conexões com AWS contas](#).

Etapa 2: conceder acesso à função IAM do CodeBuild projeto para usar a conexão

Você pode conceder acesso à função IAM do CodeBuild projeto para usar os tokens do Bitbucket fornecidos pela sua conexão.

Para conceder acesso à função IAM do CodeBuild projeto

1. Crie uma função do IAM para seu CodeBuild projeto seguindo as instruções [CodeBuild Permitir interagir com outros AWS serviços](#) para seu CodeBuild projeto.
2. Ao seguir as instruções, adicione a seguinte política do IAM ao papel CodeBuild do seu projeto para conceder acesso à conexão.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codeconnections:GetConnectionToken",
        "codeconnections:GetConnection"
      ],
      "Resource": [
        "<connection-arn>"
      ]
    }
  ]
}
```

Etapa 3: Configurar CodeBuild para usar a nova conexão

Você pode configurar uma conexão como uma credencial no nível da conta e usá-la em um projeto.

AWS Management Console

Para configurar uma conexão como uma credencial de nível de conta no AWS Management Console

1. Em Provedor de origem, escolha Bitbucket.
2. Em Credencial, siga um destes procedimentos:
 - Escolha Credencial de origem padrão para usar a credencial de origem padrão da conta e aplicá-la a todos os projetos.
 - a. Se você não estiver conectado ao Bitbucket, escolha Gerenciar credencial de origem padrão.
 - b. Para Tipo de credencial, escolha CodeConnections.
 - c. Em Conexão, escolha usar uma conexão existente ou crie uma.
 - Escolha Credencial de origem personalizada para usar uma credencial de origem personalizada para substituir as configurações padrão da conta.
 - a. Para Tipo de credencial, escolha CodeConnections.
 - b. Em Conexão, escolha usar uma conexão existente ou crie uma.

AWS CLI

Para configurar uma conexão como uma credencial de nível de conta no AWS CLI

- Abra um terminal (Linux, macOS ou Unix) ou um prompt de comando (Windows). Use o AWS CLI para executar o `import-source-credentials` comando, especificando o `--auth-type` `--server-type`, e `--token` para sua conexão.

Use o seguinte comando:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type BITBUCKET --token <connection-arn>
```

Para obter mais informações sobre como configurar vários tokens em seu CodeBuild projeto, consulte [Configurar vários tokens como credenciais no nível de origem](#).

Senha da aplicação ou token de acesso do Bitbucket

Pré-requisitos

Antes de começar, é necessário adicionar os escopos de permissão adequados para a senha da aplicação ou token de acesso do Bitbucket.

Para o Bitbucket, a senha da aplicação ou token de acesso deve ter os seguintes escopos.

- repository:read: concede acesso de leitura a todos os repositórios nos quais o usuário autorizado tem acesso.
- pullrequest:read: concede acesso de leitura a solicitações pull. Se o seu projeto tiver um webhook do Bitbucket, a senha da aplicação ou token de acesso deverá ter esse escopo.
- webhook: concede acesso para webhooks. Se o projeto tiver uma operação do webhook, a senha da aplicação ou o token de acesso deverá ter esse escopo.

Para obter mais informações, consulte [Escopos para a API REST do Bitbucket Cloud](#) e [OAuth sobre o Bitbucket Cloud](#) no site do Bitbucket.

Conectar o Bitbucket a uma senha de aplicativo (console)

Para usar o console para conectar o projeto ao Bitbucket usando uma senha de aplicação, faça o seguinte ao criar um projeto. Para mais informações, consulte [Criar um projeto de compilação \(console\)](#).

1. Em Provedor de origem, escolha Bitbucket.
2. Em Credencial, siga um destes procedimentos:
 - Escolha usar as credenciais da conta para aplicar a credencial de origem padrão da sua conta a todos os projetos.
 - a. Se você não estiver conectado ao Bitbucket, escolha Gerenciar credencial da conta.
 - b. Em Tipo de credencial, escolha Senha da aplicação.
 - Se você optar por usar credenciais de nível de conta para o Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:

- a. Se você optar por usar o Secrets Manager, poderá optar por usar uma conexão secreta existente ou criar um novo segredo e, em seguida, escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).
- b. Se você optar por usar CodeBuild, insira seu nome de usuário e senha do aplicativo Bitbucket e, em seguida, escolha Salvar.
- Selecione Usar credenciais de substituição somente para este projeto para usar uma credencial de origem personalizada para substituir as configurações de credencial da sua conta.
 - a. Na lista de credenciais preenchida, escolha uma das opções em Senha do aplicativo.
 - b. Você também pode criar um novo token de senha de aplicativo selecionando criar uma nova conexão de senha de aplicativo na descrição.

Conectar o Bitbucket a um token de acesso (console)

Para usar o console para conectar o projeto ao Bitbucket usando um token de acesso, faça o que é indicado a seguir ao criar um projeto. Para mais informações, consulte [Criar um projeto de compilação \(console\)](#).

1. Em Provedor de origem, escolha Bitbucket.
2. Em Credencial, siga um destes procedimentos:
 - Escolha usar as credenciais da conta para aplicar a credencial de origem padrão da sua conta a todos os projetos.
 - a. Se você não estiver conectado ao Bitbucket, escolha Gerenciar credencial da conta.
 - b. Em Tipo de credencial, escolha Token de acesso pessoal.
 - Se você optar por usar credenciais de nível de conta para o Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:
 - a. Se você optar por usar o Secrets Manager, poderá optar por usar uma conexão secreta existente ou criar um novo segredo e, em seguida, escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).
 - b. Se você optar por usar CodeBuild, insira seu token de acesso pessoal do Bitbucket e escolha Salvar.

- Selecione Usar credenciais de substituição somente para este projeto para usar uma credencial de origem personalizada para substituir as configurações de credencial da sua conta.
 - a. Na lista de credenciais preenchida, escolha uma das opções em Token de acesso pessoal.
 - b. Você também pode criar um novo token de acesso pessoal selecionando criar uma nova conexão de token de acesso pessoal na descrição.

Conectar o Bitbucket a uma senha da aplicação ou token de acesso (CLI)

Siga estas etapas para usar o AWS CLI para conectar seu projeto ao Bitbucket usando uma senha de aplicativo ou token de acesso. Para obter informações sobre como usar o AWS CLI with AWS CodeBuild, consulte [Referência da linha de comando](#) o.

1. Execute o comando `import-source-credentials`:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Os dados formatados JSON aparecem na saída. Copie os dados para um arquivo (por exemplo, *import-source-credentials.json*) em um local no computador local ou na instância em que o AWS CLI está instalado. Modifique os dados copiados da maneira a seguir e salve os resultados.

```
{
  "serverType": "BITBUCKET",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Substitua o seguinte:

- *server-type*: Valor obrigatório. O provedor de origem usado para essa credencial. Os valores válidos são GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB e GITLAB_SELF_MANAGED.

- ***auth-type***: Valor obrigatório. O tipo de autenticação usado para se conectar a um repositório. Os valores válidos são OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS e SECRETS_MANAGER. Pois GitHub, somente PERSONAL_ACCESS_TOKEN é permitido. BASIC_AUTH só é permitido com a senha da aplicação do Bitbucket.
 - ***should-overwrite***: Valor opcional. Defina como false para impedir a substituição das credenciais de origem do repositório. Defina como true para substituir as credenciais de origem do repositório. O valor padrão é true.
 - ***token***: Valor obrigatório. Para GitHub nosso GitHub Enterprise Server, esse é o token de acesso pessoal. Para o Bitbucket, esse é o token de acesso pessoal ou senha da aplicação. Para o tipo de autenticação CODECONNECTIONS, esse é o ARN da conexão. Para o tipo de autenticação SECRETS_MANAGER, esse é o ARN secreto.
 - ***username***: Valor opcional. Esse parâmetro é ignorado para provedores GitHub de origem do GitHub Enterprise Server.
2. Para conectar a conta com uma senha da aplicação ou um token de acesso, alterne para o diretório que contém o arquivo `import-source-credentials.json` que você salvou na etapa 1 e execute o comando `import-source-credentials` novamente.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Dados no formato JSON são exibidos na saída com um Nome de recurso da Amazon (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

 Note

Se você executar o comando `import-source-credentials` com o mesmo tipo de servidor e tipo de autenticação uma segunda vez, o token de acesso armazenado será atualizado.

Depois que sua conta estiver conectada a uma senha de aplicativo, você poderá `create-project` usá-la para criar seu CodeBuild projeto. Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).

3. Para visualizar as senhas da aplicação ou os tokens de acesso conectados, execute o comando `list-source-credentials`.

```
aws codebuild list-source-credentials
```

O objeto no formato JSON `sourceCredentialsInfos` é exibido na saída:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "BITBUCKET",
      "arn": "arn"
    }
  ]
}
```

`sourceCredentialsObject` contém uma lista de informações de credenciais de origem conectadas:

- `authType` é o tipo de autenticação usada pelas credenciais. Pode ser `OAUTH`, `BASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS` ou `SECRETS_MANAGER`.
 - `serverType` é o tipo de provedor de origem. Pode ser `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB` ou `GITLAB_SELF_MANAGED`.
 - `arn` é o ARN do token.
4. Para se desconectar de um provedor de origem e remover a senha da aplicação ou os tokens de acesso, execute o comando `delete-source-credentials` com seu ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Dados no formato JSON são retornados com um ARN das credenciais excluídas.

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Aplicativo Bitbucket OAuth

Conecte o Bitbucket usando OAuth (console)

Para usar o console para conectar seu projeto ao Bitbucket usando um OAuth aplicativo, faça o seguinte ao criar um projeto. Para mais informações, consulte [Criar um projeto de compilação \(console\)](#).

1. Em Provedor de origem, escolha Bitbucket.
2. Em Credencial, siga um destes procedimentos:
 - Escolha usar as credenciais da conta para aplicar a credencial de origem padrão da sua conta a todos os projetos.
 - a. Se você não estiver conectado ao Bitbucket, escolha Gerenciar credencial da conta.
 - b. Para Tipo de credencial, escolha OAuth aplicativo.
 - Se você optar por usar credenciais de nível de conta para o Serviço, escolha qual serviço você gostaria de usar para armazenar seu token e faça o seguinte:
 - a. Se você optar por usar o Secrets Manager, poderá optar por usar uma conexão secreta existente ou criar um novo segredo e, em seguida, escolher Salvar. Para obter mais informações sobre como criar um segredo, consulte [Criar e armazenar um token em um segredo do Secrets Manager](#).
 - b. Se você optar por usar CodeBuild, em seguida, escolher Salvar.
 - Selecione Usar credenciais de substituição somente para este projeto para usar uma credencial de origem personalizada para substituir as configurações de credencial da sua conta.
 - a. Na lista de credenciais preenchida, escolha uma das opções em OAuth app.
 - b. Você também pode criar um novo token de OAuth aplicativo selecionando criar uma nova conexão de token de aplicativo OAuth na descrição.

Para revisar seus OAuth aplicativos autorizados, navegue até [Autorizações de aplicativos](#) no Bitbucket e verifique se um aplicativo chamado AWS CodeBuild (*region*) está listado.

GitLab acesso em CodeBuild

Para GitLab, você usa uma GitLab conexão para acessar o provedor de origem.

Tópicos

- [Conecte-se CodeBuild a GitLab](#)

Conecte-se CodeBuild a GitLab

As conexões permitem que você autorize e estabeleça configurações que associem seu provedor terceirizado aos AWS recursos que você usa. Conexões de código da AWS Para associar o repositório de terceiros como origem do projeto de compilação, use uma conexão.

Para adicionar um provedor de origem GitLab autogerenciado GitLab ou um provedor de origem autogerenciado CodeBuild, você pode escolher entre:

- Use o assistente Criar projeto de compilação do CodeBuild console ou a página Editar código-fonte para escolher a opção GitLab ou provedor GitLab autogerenciado. Consulte [Crie uma conexão com GitLab \(console\)](#) para adicionar o provedor de origem. O console ajuda você a criar um recurso de conexão.
- Use a CLI para criar os recursos de conexão, consulte [Crie uma conexão com GitLab \(CLI\)](#) para criar um recurso de conexão com a CLI.

Note

É possível criar uma conexão por meio do console do Developer Tools em Configurações. Consulte [Criar uma conexão](#).

Note

Ao autorizar a instalação dessa conexão em GitLab, você concede ao nosso serviço permissões para processar seus dados acessando sua conta e pode revogar as permissões a qualquer momento desinstalando o aplicativo.

Crie uma conexão com GitLab

Esta seção descreve como se GitLab conectar CodeBuild a. Para obter mais informações sobre conexões do GitLab , consulte [Conecte-se CodeBuild a GitLab](#).

Antes de começar

- Você já deve ter criado uma conta com GitLab.

 Note

As conexões fornecem acesso somente a repositórios pertencentes à conta usada para criar e autorizar a conexão.

 Note

Você pode criar conexões com um repositório no qual você tem a função de Proprietário e GitLab, em seguida, a conexão pode ser usada com o repositório com recursos como. CodeBuild Para repositórios em grupos, você não precisa ser o proprietário do grupo.

- Para especificar uma fonte para seu projeto de compilação, você já deve ter criado um repositório no GitLab.

Tópicos

- [Crie uma conexão com GitLab \(console\)](#)
- [Crie uma conexão com GitLab \(CLI\)](#)

Crie uma conexão com GitLab (console)

Use essas etapas para usar o CodeBuild console para adicionar uma conexão ao seu projeto (repositório) no GitLab.

 Note

Em vez de criar ou usar uma conexão existente na sua conta, você pode usar uma conexão compartilhada de outra AWS conta. Para obter mais informações, consulte [Compartilhar conexões com AWS contas](#).

Para criar ou editar seu projeto de compilação

1. Faça login no CodeBuild console.
2. Escolha uma das opções a seguir.

- Selecione para criar um projeto de compilação. Siga as etapas [Criar um projeto de compilação \(console\)](#) para concluir a primeira tela e, na seção Fonte, em Provedor de origem, escolha GitLab.
 - Opte por editar um projeto de compilação existente. Escolha Editar e depois Origem. Na página Editar fonte, em Provedor de origem, escolha GitLab.
3. Escolha uma das seguintes opções:
- Em Conexão, escolha Conexão padrão. A conexão padrão aplica uma GitLab conexão padrão em todos os projetos.
 - Em Conexão, escolha Conexão personalizada. A conexão personalizada aplica uma GitLab conexão personalizada que substitui as configurações padrão da sua conta.
4. Execute um destes procedimentos:
- Em Conexão padrão ou Conexão personalizada, se você ainda não tiver criado uma conexão com seu provedor, escolha Criar uma nova GitLab conexão. Vá para a etapa 5 para criar uma conexão.
 - Em Conexão, se você já tiver criado uma conexão com seu provedor, escolha a conexão. Prosiga para a etapa 10.
- 
- Note**
- Se você fechar a janela pop-up antes de criar uma GitLab conexão, precisará atualizar a página.
5. Para criar uma conexão com um GitLab repositório, em Selecionar um provedor, escolha GitLab. Em Connection name (Nome da conexão), digite o nome da conexão que você deseja criar. Escolha Connect to GitLab.

[Developer Tools](#) > [Connections](#) > Create connection

Create a connection [Info](#)

Create GitLab connection [Info](#)

Connection name

► **Tags - optional**

Connect to GitLab

6. Quando a página de login for GitLab exibida, faça login com suas credenciais e escolha Entrar.
7. Se esta for a primeira vez que você autoriza a conexão, uma página de autorização será exibida com uma mensagem solicitando autorização para que a conexão acesse sua GitLab conta.

Escolha Authorize.

Authorize **AWS Connector for GitLab** to use your account?

An application called **AWS Connector for GitLab** is requesting access to your GitLab account. This application was created by **Amazon AWS**. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**

Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.

- **Read the authenticated user's personal information**

Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

- **Read Api**

Grants read access to the API, including all groups and projects, the container registry, and the package registry.

- **Allows read-only access to the repository**

Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.

- **Allows read-write access to the repository**

Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

8. O navegador retorna à página do console de conexões. Nas configurações de GitLab conexão, a nova conexão é mostrada em Nome da conexão.
9. Selecione Conectar.

Depois que uma GitLab conexão for criada com sucesso, um banner de sucesso será exibido na parte superior.

10. Na página Criar projeto de compilação, na lista suspensa Conexão padrão ou Conexão personalizada, verifique se o ARN da conexão está listado. Se não, escolha o botão Atualizar para que apareça.
11. No Repositório, escolha o nome do seu projeto em GitLab especificando o caminho do projeto com o namespace. Por exemplo, para um repositório em nível de grupo, insira o nome do repositório no seguinte formato: group-name/repository-name. [Para obter mais informações sobre o caminho e o namespace, consulte o path_with_namespace campo em `https://docs.gitlab.com/ee/api/projects.html#\_get-single-project`](#) Para obter mais informações sobre o namespace em GitLab, consulte <https://docs.gitlab.com/ee/user/namespace/>.

 Note

Para grupos em GitLab, você deve especificar manualmente o caminho do projeto com o namespace. Por exemplo, para um repositório myrepo em um grupo mygroup, insira o seguinte: mygroup/myrepo. Você pode encontrar o caminho do projeto com o namespace na URL em. GitLab

12. Em Versão de origem - opcional, insira um ID de solicitação pull, uma ramificação, um ID de confirmação, uma tag ou uma referência e um ID de confirmação. Para obter mais informações, consulte [Amostra da versão de origem com AWS CodeBuild](#).

 Note

Recomendamos que você escolha nomes de ramificações do Git que não se pareçam com commit IDs, como 811dd1ba1aba14473856cee38308caed7190c0d ou. 5392f7 Isso ajuda você a evitar colisões de checkout do Git com confirmações reais.

13. Em Profundidade do clone de Git - opcional, você pode criar um clone superficial com um histórico truncado para o número especificado de confirmações. Se quiser um clone completo, escolha Full.
14. Em Status da compilação - opcional, selecione Relatar status de compilação ao provedor de origem quando as compilações iniciarem e terminarem se quiser relatar os status de início e conclusão da compilação ao provedor de origem.

Para poder relatar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Crie uma conexão com GitLab (CLI)

Você pode usar o AWS Command Line Interface (AWS CLI) para criar uma conexão.

Para fazer isso, use o comando create-connection.

Important

Uma conexão criada por meio do AWS CLI ou AWS CloudFormation está no PENDING status por padrão. Depois de criar uma conexão com a CLI ou AWS CloudFormation, use o console para editar a conexão e definir seu status. AVAILABLE

Para criar uma conexão

- Siga as instruções no Guia do usuário do console Developer Tools para [Criar uma conexão com GitLab \(CLI\)](#).

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço

chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição [aws:SourceAccount](#) global [aws:SourceArn](#) e as chaves de contexto nas políticas de recursos para limitar as permissões que AWS CodeBuild concede outro serviço ao recurso. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, `.arn:aws:codebuild:*:123456789012:*`

Se o valor de `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões.

O valor de `aws:SourceArn` deve ser o ARN CodeBuild do projeto.

O exemplo a seguir mostra como você pode usar as chaves de contexto de condição `aws:SourceAccount` global `aws:SourceArn` e as chaves de contexto CodeBuild para evitar o confuso problema substituto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
        }
      }
    }
  ]
}
```

```
}  
  ]  
}
```

Tópicos avançados

Esta seção inclui vários tópicos avançados que são úteis para usuários do AWS CodeBuild mais experientes.

Tópicos

- [Permita que os usuários interajam com CodeBuild](#)
- [CodeBuild Permitir interagir com outros AWS serviços](#)
- [Criptografar saídas da compilação usando uma chave gerenciada pelo cliente](#)
- [Interaja com CodeBuild o uso do AWS CLI](#)
- [Referência de linha de comando para AWS CodeBuild](#)
- [AWS SDKs e referência de ferramentas para AWS CodeBuild](#)
- [Usando esse serviço com um AWS SDK](#)
- [Especifique o AWS CodeBuild endpoint](#)
- [Use AWS CodeBuild with AWS CodePipeline para testar o código e executar compilações](#)
- [Use AWS CodeBuild com Codecov](#)
- [Use AWS CodeBuild com Jenkins](#)
- [Use AWS CodeBuild com aplicativos sem servidor](#)
- [Avisos de terceiros AWS CodeBuild para Windows](#)
- [Use chaves de CodeBuild condição como variáveis de função de serviço do IAM para controlar o acesso à compilação](#)
- [AWS CodeBuild chaves de condição](#)

Permita que os usuários interajam com CodeBuild

Se você seguir as etapas [Conceitos básicos que usam o console](#) AWS CodeBuild para acessar pela primeira vez, provavelmente não precisará das informações deste tópico. No entanto, à medida que você continua usando CodeBuild, talvez queira fazer coisas como dar a outros usuários e grupos da sua organização a capacidade de interagir com CodeBuild.

Para permitir que um usuário ou grupo do IAM interaja com AWS CodeBuild, você deve dar a eles permissões de acesso CodeBuild a. Esta seção descreve como fazer isso pelo console do IAM ou a AWS CLI.

Se você acessar CodeBuild com sua conta AWS root (não recomendada) ou com um usuário administrador em sua AWS conta, não precisará seguir estas instruções.

Para obter informações sobre contas AWS raiz e usuários administradores, consulte [O usuário Conta da AWS raiz](#) e [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.

Para adicionar permissões de CodeBuild acesso a um grupo ou usuário do IAM (console)

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

Você já deve ter feito login no AWS Management Console usando uma das seguintes opções:

- Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [The Conta da AWS root user](#) no Guia do usuário.
- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Criando seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
- Um usuário em sua AWS conta com permissão para realizar o seguinte conjunto mínimo de ações:

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam:ListAttachedGroupPolicies
iam:ListAttachedUserPolicies
iam:ListGroups
iam:ListPolicies
iam:ListUsers
```

Para obter mais informações, consulte [Overview of IAM Policies](#) no Guia do usuário.

2. No painel de navegação, selecione Políticas.
3. Para adicionar um conjunto personalizado de permissões de AWS CodeBuild acesso a um grupo do IAM ou usuário do IAM, vá para a etapa 4 deste procedimento.

Para adicionar um conjunto padrão de permissões de CodeBuild acesso a um grupo do IAM ou usuário do IAM, escolha Tipo de política, AWS Gerenciado e faça o seguinte:

- Para adicionar permissões de acesso total ao CodeBuild, selecione a caixa chamada AWSCodeBuildAdminAccess, escolha Ações de política e, em seguida, escolha Anexar. Selecione a caixa ao lado do grupo ou do usuário do IAM de destino e, depois, escolha

Anexar política. Repita isso para as políticas chamadas AmazonS3 ReadOnlyAccess e Access. IAMFull

- Para adicionar permissões de acesso a tudo, CodeBuild exceto à administração do projeto de compilação, selecione a caixa chamada AWSCodeBuildDeveloperAccess, escolha Ações de política e, em seguida, escolha Anexar. Selecione a caixa ao lado do grupo ou do usuário do IAM de destino e, depois, escolha Anexar política. Repita isso para a política chamada AmazonS3 ReadOnlyAccess.
- Para adicionar permissões de acesso somente para leitura CodeBuild, selecione as caixas nomeadas. AWSCodeBuildReadOnlyAccess Selecione a caixa ao lado do grupo ou do usuário do IAM de destino e, depois, escolha Anexar política. Repita isso para a política chamada AmazonS3 ReadOnlyAccess.

Agora você adicionou um conjunto padrão de permissões de CodeBuild acesso a um grupo ou usuário do IAM. Ignore as etapas seguintes deste procedimento.

4. Escolha Create Policy.
5. Na página Create Policy, próximo a Create Your Own Policy, escolha Select.
6. Na página Review Policy (Revisar política), em Policy Name (Nome da política), insira um nome para a política (por exemplo, **CodeBuildAccessPolicy**). Se você usar um nome diferente, certifique-se de usá-lo durante todo este procedimento.
7. Em Policy Document (Documento de política), insira o seguinte e selecione Create Policy (Criar política):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "codebuild:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeBuildRolePolicy",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ]
    }
  ]
}
```

```
    ],
    "Resource": "arn:aws:iam::account-ID:role/role-name"
  },
  {
    "Sid": "CloudWatchLogsAccessPolicy",
    "Effect": "Allow",
    "Action": [
      "logs:FilterLogEvents",
      "logs:GetLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3AccessPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:GetObject",
      "s3:List*",
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

Note

Essa política permite o acesso a todas CodeBuild as ações e a um número potencialmente grande de AWS recursos. Para restringir as permissões a CodeBuild ações específicas, altere o valor de `codebuild:*` na declaração CodeBuild de política. Para obter mais informações, consulte [Gerenciamento de identidade e acesso](#). Para

restringir o acesso a AWS recursos específicos, altere o valor do Resource objeto. Para obter mais informações, consulte [Gerenciamento de identidade e acesso](#).

8. No painel de navegação, selecione Groups ou Users.
9. Na lista de grupos ou usuários, escolha o nome do grupo do IAM ou do usuário do IAM ao qual você deseja adicionar permissões de CodeBuild acesso.
10. Para um grupo, na página de configurações do grupo, na guia Permissions (Permissões), expanda Managed Policies (Políticas gerenciadas) e escolha Attach Policy (Anexar política).

Para um usuário, na página de configurações do usuário, na guia Permissions, escolha Add permissions.

11. Para um grupo, na página Anexar política CodeBuildAccessPolicy, selecione e escolha Anexar política.

Para um usuário, na página Adicionar permissões, escolha Anexar políticas existentes diretamente. Selecione CodeBuildAccessPolicy, escolha Avançar: Revisão e, em seguida, escolha Adicionar permissões.

Para adicionar permissões de CodeBuild acesso a um grupo ou usuário do IAM (AWS CLI)

1. Certifique-se de ter configurado o AWS CLI com a chave de AWS acesso e a chave de acesso AWS secreta que correspondem a uma das entidades do IAM, conforme descrito no procedimento anterior. Para obter mais informações, consulte [Noções básicas de configuração do AWS Command Line Interface](#) no Guia do usuário do AWS Command Line Interface .
2. Para adicionar um conjunto personalizado de permissões de AWS CodeBuild acesso a um grupo do IAM ou usuário do IAM, vá para a etapa 3 deste procedimento.

Para adicionar um conjunto padrão de permissões de CodeBuild acesso a um grupo do IAM ou usuário do IAM, faça o seguinte:

Execute os seguintes comandos, dependendo se deseja adicionar permissões a um grupo ou um usuário do IAM:

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn  
  
aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Você deve executar o comando três vezes, substituindo *group-name* ou *user-name* pelo nome do grupo ou nome de usuário do IAM e substituindo *policy-arn* uma vez por cada uma das seguintes políticas Amazon Resource Names (ARNs):

- Para adicionar permissões de acesso total ao CodeBuild, use a seguinte política ARNs:
 - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
 - `arn:aws:iam::aws:policy/IAMFullAccess`
- Para adicionar permissões de acesso a CodeBuild tudo, exceto à administração do projeto de compilação, use a seguinte política ARNs:
 - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Para adicionar permissões de acesso somente para leitura CodeBuild, use a seguinte política ARNs
 - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Agora você adicionou um conjunto padrão de permissões de CodeBuild acesso a um grupo ou usuário do IAM. Ignore as etapas seguintes deste procedimento.

3. Em um diretório vazio na estação de trabalho local ou na instância em que o AWS CLI está instalado, crie um arquivo chamado `put-group-policy.json` ou `put-user-policy.json`. Se você escolher um nome de arquivo diferente, certifique-se de usá-lo durante todo este procedimento.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "codebuild:*"
      ],
      "Resource": "*"
    }
  ],
  {
```

```
    "Sid": "CodeBuildRolePolicy",
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::account-ID:role/role-name"
},
{
    "Sid": "CloudWatchLogsAccessPolicy",
    "Effect": "Allow",
    "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
    ],
    "Resource": "*"
},
{
    "Sid": "S3AccessPolicy",
    "Effect": "Allow",
    "Action": [
        "s3:CreateBucket",
        "s3:GetObject",
        "s3:List*",
        "s3:PutObject"
    ],
    "Resource": "*"
},
{
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
    ],
    "Resource": "*"
}
]
```

Note

Essa política permite o acesso a todas CodeBuild as ações e a um número potencialmente grande de AWS recursos. Para restringir as permissões a CodeBuild

ações específicas, altere o valor de `codebuild:*` na declaração CodeBuild de política. Para obter mais informações, consulte [Gerenciamento de identidade e acesso](#). Para restringir o acesso a AWS recursos específicos, altere o valor do Resource objeto relacionado. Para obter mais informações, consulte [Gerenciamento de identidade e acesso](#) ou a documentação específica de segurança do serviço AWS .

4. Navegue até o diretório onde você salvou o arquivo e execute um dos seguintes comandos. Você pode usar valores diferentes para `CodeBuildGroupAccessPolicy` e `CodeBuildUserAccessPolicy`. Se você usar valores diferentes, certifique-se de usá-los aqui.

Para um grupo IAM:

```
aws iam put-group-policy --group-name group-name --policy-name  
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Para um usuário do :

```
aws iam put-user-policy --user-name user-name --policy-name  
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

Nos comandos anteriores, substitua *group-name* ou *user-name* pelo nome do grupo ou usuário do IAM de destino.

CodeBuild Permitir interagir com outros AWS serviços

Se você seguir as etapas [Conceitos básicos que usam o console](#) AWS CodeBuild para acessar pela primeira vez, provavelmente não precisará das informações deste tópico. No entanto, à medida que você continua usando CodeBuild, talvez queira fazer coisas como permitir CodeBuild a interação com outros AWS serviços.

Para permitir CodeBuild a interação com AWS serviços dependentes em seu nome, você precisa de uma função AWS CodeBuild de serviço. Você pode criar uma função CodeBuild de serviço usando os AWS CodePipeline consoles CodeBuild ou. Para obter mais informações, consulte:

- [Criar um projeto de compilação \(console\)](#)
- [Crie um pipeline que use CodeBuild \(CodePipelineconsole\)](#)
- [Adicionar uma ação de CodeBuild criação a um pipeline \(CodePipelineconsole\)](#)

- [Alterar configurações de um projeto de compilação \(console\)](#)

Se você não planeja usar esses consoles, esta seção descreve como criar uma função de CodeBuild serviço com o console do IAM ou o AWS CLI

 Important

CodeBuild usa a função de serviço para todas as operações que são realizadas em seu nome. Se o perfil inclui permissões que o usuário não precisa, você pode ampliar involuntariamente as permissões de um usuário. Certifique-se de que a função concede o [menor privilégio](#).

A função de serviço descrita nesta página contém uma política que concede as permissões mínimas necessárias para usar o CodeBuild. Talvez seja necessário adicionar permissões, dependendo do caso de uso.

Para criar uma função CodeBuild de serviço (console)

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

Você já deverá estar conectado no console de uma das seguintes maneiras:

- Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [The Conta da AWS root user](#) no Guia do usuário.
- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Criando seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
- Um usuário em sua AWS conta com permissão para realizar o seguinte conjunto mínimo de ações:

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam:ListAttachedRolePolicies
iam:ListPolicies
iam:ListRoles
iam:PassRole
```

```
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Para obter mais informações, consulte [Overview of IAM Policies](#) no Guia do usuário.

2. No painel de navegação, selecione Políticas.
3. Escolha Create Policy.
4. Na página Create Policy, escolha JSON.
5. Na política JSON, insira o seguinte e escolha Review Policy (Revisar política):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3PutObjectPolicy",
```

```

    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": "*"
},
{
    "Sid": "ECRPullPolicy",
    "Effect": "Allow",
    "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
    ],
    "Resource": "*"
},
{
    "Sid": "ECRAuthPolicy",
    "Effect": "Allow",
    "Action": [
        "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
},
{
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
    ],
    "Resource": "*"
}
]
}

```

Note

Essa política contém declarações que permitem o acesso a um número potencialmente grande de AWS recursos. AWS CodeBuild Para restringir o acesso a AWS recursos específicos, altere o valor da Resource matriz. Para obter mais informações, consulte a documentação de segurança do AWS serviço.

- Na página Review Policy (Revisar política), em Policy Name (Nome da política), insira um nome para a política (por exemplo, **CodeBuildServiceRolePolicy**) e escolha Create policy (Criar política).

 Note

Se você usar um nome diferente, certifique-se de usá-lo durante todo este procedimento.

- No painel de navegação, selecione Perfis.
- Escolha Criar Perfil.
- Na página Criar função, com o AWS Serviço já selecionado, escolha e, em seguida CodeBuild, escolha Avançar:Permissões.
- Na página Anexar políticas de permissões, selecione e CodeBuildServiceRolePolicy, em seguida, escolha Avançar: Revisão.
- Na página Create role and review (Criar função e revisão), em Role name (Nome da função), insira um nome para a função (por exemplo, **CodeBuildServiceRole**) e escolha Create role (Criar função).

Para criar uma função CodeBuild de serviço (AWS CLI)

- Certifique-se de ter configurado o AWS CLI com a chave de AWS acesso e a chave de acesso AWS secreta que correspondem a uma das entidades do IAM, conforme descrito no procedimento anterior. Para obter mais informações, consulte [Noções básicas de configuração do AWS Command Line Interface](#) no Guia do usuário do AWS Command Line Interface .
- Em um diretório vazio na estação de trabalho local ou na instância em que o AWS CLI está instalado, crie dois arquivos chamados `create-role.json` e `put-role-policy.json`. Se você escolher nomes de arquivo diferentes, certifique-se de usá-los durante este procedimento.

`create-role.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
```



```
    "Action": "sts:AssumeRole"
  }
]
}
```

Note

Recomendamos o uso das chaves de condição `aws:SourceAccount` e `aws:SourceArn` para se proteger contra o [problema confused deputy](#). Por exemplo, é possível editar a política de confiança anterior com os blocos de condição a seguir. `aws:SourceAccount` É o proprietário do CodeBuild projeto e o `aws:SourceArn` ARN do CodeBuild projeto.

Se você quiser restringir sua função de serviço a uma AWS conta, `create-role.json` pode ter uma aparência semelhante a esta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "account-ID"
          ]
        }
      }
    }
  ]
}
```

Se você quiser restringir sua função de serviço a um CodeBuild projeto específico, `create-role.json` pode ter uma aparência semelhante a esta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
        }
      }
    }
  ]
}
```

Note

Se você não sabe ou não escolheu um nome para seu CodeBuild projeto e quer uma restrição de política de confiança em um determinado padrão de ARN, você pode substituir essa parte do ARN por um caractere curinga (*). Depois de criar o projeto, você poderá atualizar a política de confiança.

put-role-policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    }
  ],
}
```

```
    "Resource": "*"
  },
  {
    "Sid": "CodeCommitPolicy",
    "Effect": "Allow",
    "Action": [
      "codecommit:GitPull"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3GetObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3PutObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

Note

Essa política contém declarações que permitem o acesso a um número potencialmente grande de AWS recursos. AWS CodeBuild Para restringir o acesso a AWS recursos

específicos, altere o valor da Resource matriz. Para obter mais informações, consulte a documentação de segurança do AWS serviço.

3. Navegue até o diretório onde você salvou os arquivos anteriores, e execute os dois comandos seguintes, um por vez, nessa ordem. Você pode usar valores diferentes para CodeBuildServiceRole e CodeBuildServiceRolePolicy, mas não se esqueça de usá-los aqui.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document  
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name  
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

Criptografar saídas da compilação usando uma chave gerenciada pelo cliente

Se você seguir as etapas [Conceitos básicos que usam o console](#) AWS CodeBuild para acessar pela primeira vez, provavelmente não precisará das informações deste tópico. No entanto, à medida que você continua usando CodeBuild, talvez queira fazer coisas como criptografar artefatos de construção.

Para AWS CodeBuild criptografar seus artefatos de saída de compilação, ele precisa acessar uma chave KMS. Por padrão, CodeBuild usa o Chave gerenciada pela AWS para Amazon S3 em sua AWS conta.

Se você não quiser usar o Chave gerenciada pela AWS, você mesmo deverá criar e configurar uma chave gerenciada pelo cliente. Esta seção descreve como fazer isso pelo console do IAM.

Para obter informações sobre chaves gerenciadas pelo cliente, consulte [AWS Key Management Service Concepts](#) e [Creating Keys](#) no Guia do desenvolvedor do AWS KMS .

Para configurar uma chave gerenciada pelo cliente para uso por CodeBuild, siga as instruções na seção “Como modificar uma política de chaves” de [Modificar uma política de chaves](#) no Guia do AWS KMS desenvolvedor. Em seguida, adicione as seguintes declarações (entre **### BEGIN ADDING STATEMENTS HERE ###** e **### END ADDING STATEMENTS HERE ###**) à política

principal. As elipses (...) são usadas para agilizar e para ajudá-lo a encontrar onde adicionar as declarações. Não remova nenhuma declaração e não digite essas elipses nas políticas de chaves.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENTS HERE ###
    {
      "Sid": "Allow access through Amazon S3 for all principals in the account that are
authorized to use Amazon S3",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.region-ID.amazonaws.com",
          "kms:CallerAccount": "account-ID"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-ID:role/CodeBuild-service-role"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    },  
    ### END ADDING STATEMENTS HERE ###  
    {  
        "Sid": "Enable IAM User Permissions",  
        ...  
    },  
    {  
        "Sid": "Allow access for Key Administrators",  
        ...  
    },  
    {  
        "Sid": "Allow use of the key",  
        ...  
    },  
    {  
        "Sid": "Allow attachment of persistent resources",  
        ...  
    }  
]  
}
```

- ***region-ID*** representa o ID da AWS região em que os buckets do Amazon S3 associados CodeBuild estão localizados (por exemplo,). `us-east-1`
- ***account-ID*** representa o ID da AWS conta que possui a chave gerenciada pelo cliente.
- ***CodeBuild-service-role*** representa o nome da função de CodeBuild serviço que você criou ou identificou anteriormente neste tópico.

Note

Para criar ou configurar uma chave gerenciada pelo cliente por meio do console do IAM, você deve primeiro fazer login no AWS Management Console usando uma das seguintes opções:

- Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [O usuário raiz da conta](#) no Guia do usuário do .
- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Criando seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.

- Um usuário em sua AWS conta com permissão para criar ou modificar a chave gerenciada pelo cliente. Para obter mais informações, consulte [Permissões necessárias para usar o AWS KMS console](#) no Guia do AWS KMS desenvolvedor.

Interaja com CodeBuild o uso do AWS CLI

Se você seguir as etapas [Conceitos básicos que usam o console](#) AWS CodeBuild para acessar pela primeira vez, provavelmente não precisará das informações deste tópico. No entanto, à medida que você continua usando CodeBuild, talvez queira fazer coisas como permitir que os usuários usem o AWS CLI para interagir em CodeBuild vez de (ou além de) o CodeBuild CodePipeline console, o console ou AWS SDKs o.

Para instalar e configurar o AWS CLI, consulte [Como configurar o AWS Command Line Interface](#) no Guia do AWS Command Line Interface usuário.

Depois de instalar o AWS CLI, conclua as seguintes tarefas:

1. Execute o comando a seguir para confirmar se a instalação dos AWS CLI suportes CodeBuild:

```
aws codebuild list-builds
```

Se bem-sucedido, informações semelhantes a estas serão exibidas como resultado:

```
{
  "ids": []
}
```

Os colchetes vazios indicam que você ainda não executou nenhuma compilação.

2. Se for exibido um erro, você deve desinstalar a versão atual do AWS CLI e, então, instalar a versão mais recente. Para obter mais informações, consulte [Desinstalação da AWS CLI](#) e [Instalação da AWS Command Line Interface](#) no Guia do usuário do AWS Command Line Interface .

Referência de linha de comando para AWS CodeBuild

AWS CLI Ele fornece comandos para automatização AWS CodeBuild. Use as informações neste tópico como um suplemento ao [Guia do usuário do AWS Command Line Interface](#) e à [Referência da AWS CLI para AWS CodeBuild](#).

Não é o que você está procurando? Se você quiser usar o AWS SDKs para ligar CodeBuild, veja [AWS SDKs e referência de ferramentas](#) o.

Para usar as informações deste tópico, você já deve ter instalado o AWS CLI e configurado para uso com CodeBuild, conforme descrito em [Interaja com CodeBuild o uso do AWS CLI](#).

Para usar o AWS CLI para especificar o endpoint para CodeBuild, consulte [Especifique o AWS CodeBuild endpoint \(AWS CLI\)](#).

Execute esse comando para obter uma lista de CodeBuild comandos.

```
aws codebuild help
```

Execute esse comando para obter informações sobre um CodeBuild comando, onde *command-name* está o nome do comando.

```
aws codebuild command-name help
```

CodeBuild os comandos incluem:

- `batch-delete-builds`: exclui uma ou mais incorporações. CodeBuild Para obter mais informações, consulte [Excluir compilações \(AWS CLI\)](#).
- `batch-get-builds`: obtém informações sobre várias compilações no CodeBuild. Para obter mais informações, consulte [Visualizar detalhes de compilação \(AWS CLI\)](#).
- `batch-get-projects`: Obter informações sobre um ou mais projetos de build especificados. Para obter mais informações, consulte [Visualizar detalhes de um projeto de compilação \(AWS CLI\)](#).
- `create-project`: Cria um projeto de build. Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#).
- `delete-project`: Exclui um projeto de build. Para obter mais informações, consulte [Excluir um projeto de compilação \(AWS CLI\)](#).

- `list-builds`: Lista os nomes de recursos da Amazon (ARNs) para compilações. CodeBuild Para obter mais informações, consulte [Veja uma lista de build IDs \(AWS CLI\)](#).
- `list-builds-for-project`: obtém uma lista de IDs compilações associadas a um projeto de compilação especificado. Para obter mais informações, consulte [Exibir uma lista de IDs compilações para um projeto de construção \(AWS CLI\)](#).
- `list-curated-environment-images`: obtém uma lista de imagens do Docker gerenciadas pelas CodeBuild quais você pode usar em suas compilações. Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#).
- `list-projects`: Obter uma lista de nomes de projeto de build. Para obter mais informações, consulte [Visualizar uma lista de nomes de projetos de compilação \(AWS CLI\)](#).
- `start-build`: Começa a execução de um build. Para obter mais informações, consulte [Executar uma compilação \(AWS CLI\)](#).
- `stop-build`: Tentar parar a execução de um build especificado. Para obter mais informações, consulte [Parar uma compilação \(AWS CLI\)](#).
- `update-project`: Alterar informações sobre o projeto de build especificado. Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(AWS CLI\)](#).

AWS SDKs e referência de ferramentas para AWS CodeBuild

Para usar uma das ferramentas AWS SDKs ou para automatizar AWS CodeBuild, consulte os recursos a seguir.

Se você quiser usar o AWS CLI para executar CodeBuild, consulte [Referência da linha de comando](#) o.

Suporte AWS SDKs e ferramentas para AWS CodeBuild

O seguinte AWS SDKs e as ferramentas suportam CodeBuild:

- O [AWS SDK para C++](#). Para obter mais informações, consulte a seção de CodeBuild namespace [Aws::](#) da referência da API AWS SDK for C++.
- O [AWS SDK para Go](#). Para obter mais informações, consulte a seção [codebuild](#) da Referência da API do AWS SDK para Go.
- O [AWS SDK para Java](#). Para obter mais informações, consulte as seções `com.amazonaws.services.codebuild` e `com.amazonaws.services.codebuild.model` da [Referência do AWS SDK para Java API](#).

- O [AWS SDK para JavaScript o navegador](#) e o [AWS SDK para JavaScript](#) o Node.js. Para maiores informações, veja a [Classe: AWS. CodeBuild](#) seção do AWS SDK para referência de JavaScript API.
- O [AWS SDK para .NET](#). Para obter mais informações, consulte as seções do namespace [Amazon.CodeBuild](#) e [Amazon.CodeBuild.Model](#) do AWS Referência do SDK para API .NET.
- O [AWS SDK para PHP](#). Para obter mais informações, consulte a seção [Namespace Aws \CodeBuild](#) da AWS Referência do SDK para API PHP.
- O [AWS SDK para Python \(Boto3\)](#). Para obter mais informações, consulte a seção [CodeBuild](#) da Boto 3 Documentation.
- O [AWS SDK para Ruby](#). Para obter mais informações, consulte a seção [Module: Aws::CodeBuild](#) da AWS Referência do SDK para API Ruby.
- As [AWS ferramentas para PowerShell](#). Para obter mais informações, consulte a [AWS CodeBuild](#) seção AWS Tools for PowerShell Cmdlet Reference.

Usando esse serviço com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código

Documentação do SDK	Exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Para obter exemplos específicos deste serviço, consulte [Exemplos de código para CodeBuild usar AWS SDKs](#).

Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link [Fornecer feedback na parte inferior desta página](#).

Especifique o AWS CodeBuild endpoint

Você pode usar o AWS Command Line Interface (AWS CLI) ou um dos AWS SDKs para especificar o endpoint usado por AWS CodeBuild. Há um endpoint para cada região em que CodeBuild está disponível. Além de um endpoint regional, quatro regiões também têm um endpoint do Federal Information Processing Standard (FIPS). Para obter mais informações sobre os endpoints FIPS, consulte [Visão geral da publicação 140-2 da FIPS](#).

A especificação de um endpoint é opcional. Se você não informar explicitamente CodeBuild qual endpoint usar, o serviço usará o endpoint associado à região que sua AWS conta usa. CodeBuild nunca usa como padrão um endpoint FIPS. Se você quiser usar um endpoint do FIPS, deverá associar o CodeBuild a ele usando um dos métodos a seguir.

 Note

Você pode usar um alias ou nome de região para especificar um endpoint usando um AWS SDK. Se você usar o AWS CLI, deverá usar o nome completo do endpoint.

Para endpoints que podem ser usados com CodeBuild, consulte [CodeBuild regiões e endpoints](#).

Tópicos

- [Especifique o AWS CodeBuild endpoint \(AWS CLI\)](#)
- [Especifique o AWS CodeBuild endpoint \(AWS SDK\)](#)

Especifique o AWS CodeBuild endpoint (AWS CLI)

Você pode usar o AWS CLI para especificar o endpoint pelo qual AWS CodeBuild é acessado usando o `--endpoint-url` argumento em qualquer CodeBuild comando. Por exemplo, execute esse comando para obter uma lista de nomes de compilação de projetos usando o endpoint do Federal Information Processing Standards (FIPS) na região do Leste dos EUA (Norte da Virgínia):

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Inclua `https://` no início do endpoint.

O `--endpoint-url` AWS CLI argumento está disponível para todos os AWS serviços. Para obter mais informações sobre esse e outros AWS CLI argumentos, consulte [Referência de AWS CLI comandos](#).

Especifique o AWS CodeBuild endpoint (AWS SDK)

Você pode usar um AWS SDK para especificar o endpoint pelo qual AWS CodeBuild é acessado. Embora esse exemplo use o [AWS SDK for Java](#), você pode especificar o endpoint com o outro. AWS SDKs

Use o `withEndpointConfiguration` método ao construir o cliente `AWSCodeBuild`. Este é o formato a ser usado:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
    "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para obter informações sobre `AWSCodeBuildClientBuilder`, consulte [Classe AWSCodeBuildClientBuilder](#).

As credenciais usadas em `withCredentials` devem ser do tipo `AWSStaticCredentialsProvider`. Para obter mais informações, consulte [Trabalhando com AWS credenciais](#).

Não inclua `https://` no início do endpoint.

Se você quiser especificar um endpoint diferente de FIPS, poderá usar a região, em vez do próprio endpoint. Por exemplo, para especificar o endpoint na região do Leste dos EUA (Norte da Virgínia), é possível usar `us-east-1` em vez do nome completo do endpoint, `codebuild.us-east-1.amazonaws.com`.

Se você quiser especificar um endpoint do FIPS, use um alias para simplificar o código. Somente endpoints do FIPS têm um alias. Outros endpoints devem ser especificados usando a região ou o nome completo.

A seguinte tabela indica o alias de cada um dos quatro endpoints do FIPS disponíveis:

Nome da região	Região	Endpoint	Alias
Leste dos EUA (Norte da Virgínia)	us-east-1	codebuild-fips.us-east-1.amazonaws.com	us-east-1-fips
Leste dos EUA (Ohio)	us-east-2	codebuild-fips.us-east-2.amazonaws.com	us-east-2-fips
Oeste dos EUA (N. da Califórnia)	us-west-1	codebuild-fips.us-west-1.amazonaws.com	us-west-1-fips

Nome da região	Região	Endpoint	Alias
Oeste dos EUA (Oregon)	us-west-2	codebuild-fips.us-west-2.amazonaws.com	us-west-2-fips

Para especificar o uso do endpoint do FIPS na região do Oeste dos EUA (Oregon) usando um alias:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para especificar o uso do endpoint diferente do FIPS na região do Leste dos EUA (Norte da Virgínia):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
"us-east-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para especificar o uso do endpoint diferente do FIPS na região da Ásia-Pacífico (Mumbai):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
"ap-south-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Use AWS CodeBuild with AWS CodePipeline para testar o código e executar compilações

Você pode automatizar seu processo de lançamento usando AWS CodePipeline para testar seu código e executar suas compilações com AWS CodeBuild.

A tabela a seguir lista tarefas e os métodos disponíveis para executá-los. Usar o AWS SDKs para realizar essas tarefas está fora do escopo deste tópico.

Tarefa	Abordagens disponíveis	Abordagens descritas neste tópico
Crie um pipeline de entrega contínua (CD) CodePipeline que automatize as compilações com CodeBuild	• CodePipeline console • AWS CLI • AWS SDKs	• Usar o console do CodePipeline • Usar a AWS CLI • Você pode adaptar as informações neste tópico para usar AWS SDKs o. Para obter mais informações, consulte a documentação de <code>create-pipeline</code> ação da sua linguagem de programação na SDKs seção Ferramentas para Amazon Web Services ou consulte CreatePipeline a Referência da AWS CodePipeline API.
Adicione automação de teste e construção o CodeBuild a um pipeline existente no CodePipeline	• CodePipeline console • AWS CLI • AWS SDKs	• Usar o console do CodePipeline para adicionar automação de compilação • Usar o console do CodePipeline para adicionar automação de teste • Para o AWS CLI, você pode adaptar as informações neste tópico para criar um pipeline que contenha uma ação de CodeBuild criação ou uma ação de teste. Para obter mais informações, consulte Editar um pipeline (AWS CLI) e a referência da estrutura do CodePipeline pipeline no Guia AWS CodePipeline do usuário. • Você pode adaptar as informações neste tópico para usar AWS SDKs o. Para obter mais informações, consulte a documentação de <code>update-pipeline</code> ação da sua linguagem de programação na

Tarefa	Abordagens disponíveis	Abordagens descritas neste tópico
		SDKs seção Ferramentas para Amazon Web Services ou consulte UpdatePipeline a Referência da AWS CodePipeline API.

Tópicos

- [Pré-requisitos](#)
- [Crie um pipeline que use CodeBuild \(CodePipelineconsole\)](#)
- [Criar um pipeline que use o CodeBuild \(AWS CLI\)](#)
- [Adicionar uma ação de CodeBuild criação a um pipeline \(CodePipelineconsole\)](#)
- [Adicionar uma ação CodeBuild de teste a um pipeline \(CodePipeline console\)](#)

Pré-requisitos

1. Responda às perguntas em [Planejar uma compilação](#).
2. Se você estiver usando um usuário para acessar CodePipeline em vez de uma conta AWS raiz ou um usuário administrador, anexe a política gerenciada nomeada `AWSCodePipelineFullAccess` ao usuário (ou ao grupo do IAM ao qual o usuário pertence). Não é recomendável usar uma conta AWS root. Essa política concede ao usuário permissão para criar o pipeline no CodePipeline. Para obter mais informações, consulte [Attaching managed policies](#) no Guia do usuário.

Note

A entidade do IAM que associa a política ao usuário (ou ao grupo ao qual o usuário pertence) deve ter permissão no IAM para associar políticas. Para obter mais informações, consulte [Delegating permissions to administer IAM users, groups, and credentials](#) no Guia do usuário.

3. Crie uma função de CodePipeline serviço, caso ainda não tenha uma disponível em sua AWS conta. CodePipeline usa essa função de serviço para interagir com outros AWS serviços AWS CodeBuild, inclusive em seu nome. Por exemplo, para usar o AWS CLI para criar uma função CodePipeline de serviço, execute o `create-role` comando IAM:

Para Linux, macOS ou Unix:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role
--assume-role-policy-document '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Principal":
{"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}}'
```

Para Windows:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-
role-policy-document "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":
\"Allow\",\"Principal\":{\"Service\":\"codepipeline.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}}"
```

 Note

A entidade do IAM que cria essa função CodePipeline de serviço deve ter permissão no IAM para criar funções de serviço.

- Depois de criar uma função de CodePipeline serviço ou identificar uma existente, você deve adicionar a política de função de CodePipeline serviço padrão à função de serviço conforme descrito em [Revise a política de função de CodePipeline serviço padrão](#) no Guia AWS CodePipeline do usuário, se ela ainda não fizer parte da política da função.

 Note

A entidade do IAM que adiciona essa política CodePipeline de função de serviço deve ter permissão no IAM para adicionar políticas de função de serviço às funções de serviço.

- Crie e carregue o código-fonte em um tipo de repositório compatível com CodeBuild e CodePipeline, por exemplo CodeCommit, Amazon S3, Bitbucket ou GitHub. O código-fonte deve conter um arquivo de buildspec, mas você poderá declarar um ao definir um projeto de compilação mais adiante neste tópico. Para obter mais informações, consulte [Referência de buildspec](#).

⚠ Important

Se pretende usar o pipeline para implantar código-fonte da compilação, o artefato de saída da compilação deve ser compatível com o sistema de implantação que você usa.

- Para AWS OpsWorks, consulte [Fonte do aplicativo](#) e [Uso CodePipeline com AWS OpsWorks](#) no Guia AWS OpsWorks do usuário.

Crie um pipeline que use CodeBuild (CodePipelineconsole)

Use o procedimento a seguir para criar um pipeline usado CodeBuild para criar e implantar seu código-fonte.

Para criar um pipeline que somente teste o código-fonte:

- Use o seguinte procedimento para criar o pipeline e, então, exclua do pipeline os estágios Build e Beta. Depois use o procedimento [Adicionar uma ação CodeBuild de teste a um pipeline \(CodePipeline console\)](#) neste tópico para adicionar ao pipeline uma ação de teste que use CodeBuild.
- Use um dos outros procedimentos neste tópico para criar o pipeline e, depois, use o procedimento [Adicionar uma ação CodeBuild de teste a um pipeline \(CodePipeline console\)](#) neste tópico para adicionar ao pipeline uma ação de teste que use CodeBuild.

Para usar o assistente de criação de pipeline CodePipeline para criar um pipeline que usa CodeBuild

1. Faça login no AWS Management Console usando:

- Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [O usuário raiz da conta](#) no Guia do usuário do .
- Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
- Um usuário em sua AWS conta com permissão para usar o seguinte conjunto mínimo de ações:

```
codepipeline:*  
iam:ListRoles
```

```
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abra o AWS CodePipeline console em <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. No seletor de AWS região, escolha a AWS região em que os AWS recursos do seu projeto de construção estão localizados. Essa deve ser uma AWS região CodeBuild com suporte. Para ter mais informações, consulte [AWS CodeBuild](#) no Referência geral da Amazon Web Services.
4. Crie um pipeline. Se uma página de CodePipeline informações for exibida, escolha Criar pipeline. Se uma página All Pipelines (Todos os pipelines) for exibida, selecione Create pipeline (Criar pipeline).
5. Na página Step 1: Choose pipeline settings (Etapa 1: selecionar as configurações do pipeline), em Pipeline name (Nome do pipeline), insira o nome do pipeline (por exemplo, **CodeBuildDemoPipeline**). Se você escolher um nome diferente, certifique-se de usá-lo durante todo este procedimento.
6. Em Role name (Nome da função), siga um dos seguintes procedimentos:

Selecione New service role (Nova função de serviço), e em Role Name (Nome da função), insira o nome para a nova função de serviço.

Selecione Existing service role (Função de serviço existente) e depois selecione a função de serviço do CodePipeline criada ou identificada como parte dos pré-requisitos deste tópico.

7. Em Artifact store (Armazenamento de artefatos), siga um dos seguintes procedimentos:

- Escolha o local padrão para usar o armazenamento de artefatos padrão, como o bucket de artefatos S3 designado como padrão, para seu pipeline na AWS região que você selecionou para seu pipeline.
- Escolha um local personalizado se você já tiver um armazenamento de artefatos criado por você, como um bucket de artefatos do S3, na mesma AWS região do seu pipeline.

 Note

Este não é o bucket de origem para o código-fonte do pipeline. Este é o armazenamento de artefatos para o pipeline. Um armazenamento de artefatos separado, como um bucket do S3, é necessário para cada pipeline, na mesma AWS região do pipeline.

8. Escolha Próximo.

9. Na página Step 2: Add source stage (Etapa 2: adicionar estágio de origem), para Source provider (Fornecedor de origem), faça um dos seguintes procedimentos:

- Se o código-fonte estiver armazenado no bucket do S3, escolha Amazon S3. Em Bucket, selecione o bucket do S3 que contenha o código-fonte. Em S3 object key (Chave do objeto do S3), insira o nome do arquivo o contém o código-fonte (por exemplo *file-name.zip*). Escolha Próximo.
- Se seu código-fonte estiver armazenado em um AWS CodeCommit repositório, escolha CodeCommit. Para Repository name (Nome do repositório), escolha o nome do repositório que contém o código-fonte. Em Branch name (Nome da ramificação), escolha o nome da ramificação que contenha a versão do código-fonte que você deseja compilar. Escolha Próximo.
- Se seu código-fonte estiver armazenado em um GitHub repositório, escolha GitHub. Escolha Connect GitHub to e siga as instruções para se autenticar com GitHub. Para Repository (Repositório), escolha o nome do repositório que contém o código-fonte. Em Branch (Ramificação), escolha o nome da ramificação que contenha a versão do código-fonte que você deseja compilar.

Escolha Próximo.

10. Na página Step 3: Add build stage (Etapa 3: adicionar estágio de compilação), em Build provider (Provedor de compilação), escolha CodeBuild.

11. Se você já tiver um projeto de compilação que deseja usar, em Nome do projeto, escolha o nome do projeto de compilação e avance para a próxima etapa deste procedimento.

Se você precisar criar um novo projeto de CodeBuild compilação, siga as instruções [Criar um projeto de compilação \(console\)](#) e retorne a esse procedimento.

Se você escolher um projeto de construção existente, ele deverá ter configurações de artefato de saída de construção já definidas (mesmo que as CodePipeline substitua). Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#).

 Important

Se você habilitar webhooks para um CodeBuild projeto e o projeto for usado como uma etapa de compilação CodePipeline, duas compilações idênticas serão criadas para cada confirmação. Uma compilação é acionada por meio de webhooks e outra por meio do CodePipeline. Como o faturamento é por compilação, você é cobrado por ambas as compilações. Portanto, se você estiver usando CodePipeline, recomendamos que você desative os webhooks em. CodeBuild No console do AWS CodeBuild desmarque a caixa de Webhook. Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#).

12. Na página Step 4: Add deploy stage (Etapa 4: adicionar estágio de implantação), faça o seguinte:
 - Se você não deseja implantar o artefato de saída da compilação, selecione Skip (Avançar) e confirme essa opção quando solicitado.
 - Se quiser implantar o artefato de saída da compilação, em Deploy provider (Provedor de implantação), selecione um provedor de implantação e especifique as configurações quando solicitadas.

Escolha Próximo.

13. Na página Review (Revisar), revise suas escolhas e, em seguida, selecione Create pipeline (Criar pipeline).
14. Depois de o pipeline ser executado com sucesso, você pode obter o artefato de saída de build. Com o pipeline exibido no CodePipeline console, na ação Criar, escolha a dica de ferramenta. Anote o valor do artefato de saída (por exemplo, MyAppBuild).

 Note

Você também pode obter o artefato de saída da compilação escolhendo o link Construir artefatos na página de detalhes da compilação no CodeBuild console. Para exibir essa página, despreze as etapas restantes deste procedimento e consulte [Visualizar detalhes de compilação \(console\)](#).

15. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
16. Na lista de bucket, abra o bucket usado pelo pipeline. O nome do bucket deve seguir o formato codepipeline-*region-ID-random-number*. Você pode usar o AWS CLI para executar o CodePipeline get-pipeline comando para obter o nome do bucket, onde *my-pipeline-name* está o nome de exibição do seu pipeline:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Na saída, o objeto pipeline contém um objeto artifactStore, que contém um valor location com o nome do bucket.

17. Abra a pasta que corresponda ao nome de seu pipeline (dependendo do tamanho do nome do pipeline, o nome da pasta pode estar truncado) e abra a pasta correspondente ao valor de Output artifact (Artefato de saída) que você anotou anteriormente.
18. Extraia o conteúdo do arquivo. Se houver vários arquivos na pasta, extraia o conteúdo do arquivo com o carimbo Última modificação mais recente. (Talvez você precise atribuir ao arquivo a extensão .zip, de maneira que possa trabalhar com ele no utilitário ZIP do sistema.) O artefato de saída de compilação está no conteúdo extraído do arquivo.
19. Se você tiver instruído CodePipeline a implantar o artefato de saída de compilação, use as instruções do provedor de implantação para acessar o artefato de saída de compilação nos destinos de implantação.

Criar um pipeline que use o CodeBuild (AWS CLI)

Use o procedimento a seguir para criar um pipeline usado CodeBuild para criar seu código-fonte.

Para usar o AWS CLI para criar um pipeline que implante seu código-fonte construído ou que apenas teste seu código-fonte, você pode adaptar as instruções em [Editar um pipeline \(AWS CLI\)](#) e a [referência da estrutura do CodePipeline pipeline](#) no Guia do AWS CodePipeline usuário.

1. Crie ou identifique um projeto de construção em CodeBuild. Para obter mais informações, consulte [Criar um projeto de compilação](#).

 Important

O projeto de build deve definir as configurações de artefato de saída de build (mesmo que o CodePipeline sobrescreva-as). Para obter mais informações, consulte a descrição de artifacts em [Criar um projeto de compilação \(AWS CLI\)](#).

2. Certifique-se de ter configurado o AWS CLI com a chave de AWS acesso e a chave de acesso AWS secreta que correspondem a uma das entidades do IAM descritas neste tópico. Para obter mais informações, consulte [Noções básicas de configuração da AWS Command Line Interface](#) no Guia do usuário da AWS Command Line Interface .
3. Crie um arquivo formatado JSON que represente a estrutura do pipeline. Nomeie o arquivo como `create-pipeline.json` ou algo similar. Por exemplo, essa estrutura em formato JSON cria um pipeline com uma ação de origem que faz referência a um bucket de entrada do S3 e uma ação de compilação que use o CodeBuild:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-name>",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
```

```
        "S3Bucket": "<bucket-name>",
        "S3ObjectKey": "<source-code-file-name.zip>"
    },
    "runOrder": 1
}
]
},
{
    "name": "Build",
    "actions": [
        {
            "inputArtifacts": [
                {
                    "name": "MyApp"
                }
            ],
            "name": "Build",
            "actionTypeId": {
                "category": "Build",
                "owner": "AWS",
                "version": "1",
                "provider": "CodeBuild"
            },
            "outputArtifacts": [
                {
                    "name": "default"
                }
            ],
            "configuration": {
                "ProjectName": "<build-project-name>"
            },
            "runOrder": 1
        }
    ]
}
],
"artifactStore": {
    "type": "S3",
    "location": "<CodePipeline-internal-bucket-name>"
},
"name": "<my-pipeline-name>",
"version": 1
}
```



```
}
```

Nesses dados formatados JSON:

- O valor de `roleArn` deve corresponder ao ARN da função de CodePipeline serviço que você criou ou identificou como parte dos pré-requisitos.
- Os valores de `S3Bucket` e `S3ObjectKey` na `configuration` presumem que o código-fonte esteja armazenado no bucket do S3. Para configurações de outros tipos de repositório de código-fonte, consulte [Referência da estrutura do pipeline do CodePipeline](#) no Guia do usuário do AWS CodePipeline .
- O valor de `ProjectName` é o nome do projeto de CodeBuild construção que você criou anteriormente neste procedimento.
- O valor de `location` é o nome do bucket S3 usado por este pipeline. Para obter mais informações, consulte [Criar uma política para bucket do S3 para usar como armazenamento de artefatos do CodePipeline](#) no Guia do usuário do AWS CodePipeline .
- O valor de `name` é o nome deste pipeline. Todos os nomes de pipeline devem ser únicos em sua conta.

Embora esses dados descrevam somente uma ação de origem e uma ação de construção, você pode adicionar ações para atividades relacionadas a testes, implantação do artefato de saída de compilação, invocação de AWS Lambda funções e muito mais. Para obter mais informações, consulte [Referência da estrutura do pipeline do AWS CodePipeline](#) no Guia do usuário do AWS CodePipeline .

4. Alterne para a pasta que contém o arquivo JSON e execute o CodePipeline [create-pipeline](#) comando, especificando o nome do arquivo:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

Note

Você deve criar o pipeline em uma AWS região com suporte. CodeBuild Para ter mais informações, consulte [AWS CodeBuild](#) no Referência geral da Amazon Web Services.

Os dados formatados em JSON aparecem na saída e CodePipeline criam o pipeline.

5. Para obter informações sobre o status do pipeline, execute o CodePipeline [get-pipeline-state](#) comando, especificando o nome do pipeline:

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Na saída, procure informações que confirme que o build foi executado com sucesso. As elipses (...) são usadas para exibir dados que foram omitidos para agilizar.

```
{
  ...
  "stageStates": [
    ...
    {
      "actionStates": [
        {
          "actionName": "CodeBuild",
          "latestExecution": {
            "status": "SUCCEEDED",
            ...
          },
          ...
        }
      ]
    }
  ]
}
```

Se você executar esse comando muito cedo, poderá não ver as informações sobre a ação de build. Você pode precisar executar esse comando várias vezes até que o pipeline tenha concluído a execução da ação de build.

6. Depois de um build bem-sucedido, siga estas instruções para obter o artefato de saída de build. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

Note

Você também pode obter o artefato de saída de compilação selecionando o link Build artifacts (Criar artefatos) na página de detalhes da compilação relacionada, no console do CodeBuild. Para exibir essa página, despreze as etapas restantes deste procedimento e consulte [Visualizar detalhes de compilação \(console\)](#).

7. Na lista de bucket, abra o bucket usado pelo pipeline. O nome do bucket deve seguir o formato `codepipeline-<region-ID>-<random-number>`. Você pode obter o nome do bucket do `create-pipeline.json` arquivo ou executar o CodePipeline `get-pipeline` comando para obter o nome do bucket.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

Na saída, o objeto `pipeline` contém um objeto `artifactStore`, que contém um valor `location` com o nome do bucket.

8. Abra a pasta que corresponda ao nome do seu pipeline (por exemplo, *<pipeline-name>*).
9. Naquela pasta, abra a pasta denominada `default`.
10. Extraia o conteúdo do arquivo. Se houver vários arquivos na pasta, extraia o conteúdo do arquivo com o carimbo Última modificação mais recente. (Talvez você precise atribuir ao arquivo uma extensão `.zip`, de maneira que possa trabalhar com ele no utilitário ZIP do sistema.) O artefato de saída de compilação está no conteúdo extraído do arquivo.

Adicionar uma ação de CodeBuild criação a um pipeline (CodePipelineconsole)

1. Faça login no AWS Management Console usando:
 - Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [O usuário raiz da conta](#) no Guia do usuário do .
 - Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
 - Um usuário em sua AWS conta com permissão para realizar o seguinte conjunto mínimo de ações:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
```

```
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abra o CodePipeline console em <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. No seletor de AWS região, escolha a AWS região em que seu funil está localizado. Essa deve ser uma região CodeBuild com suporte. Para ter mais informações, consulte [CodeBuild](#) no Referência geral da Amazon Web Services.
4. Na página Pipelines, selecione o nome do pipeline.
5. Na página de detalhes do pipeline, na ação Source (Origem), selecione a dica de ferramenta. Anote o valor do artefato de saída (por exemplo, MyApp).

 Note

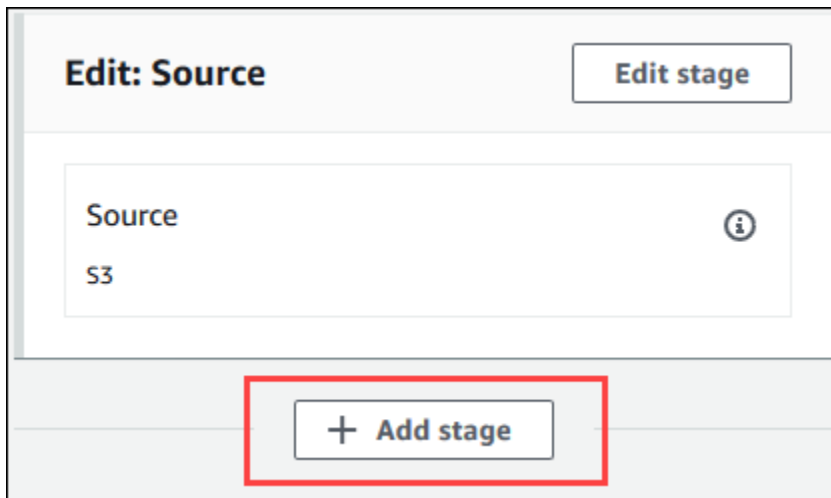
Este procedimento mostra como adicionar uma ação de build em um estágio de build entre os estágios Source (Origem) e Beta (Beta). Se quiser adicionar a ação de build a outro lugar, escolha a dica de ferramenta sobre a ação próximo ao local onde você deseja adicionar a ação de build e anote o valor para Output artifact (Artefato de saída).

6. Selecione Editar.
7. Entre os estágios Source (Origem) e Beta, selecione Add stage (Adicionar estágio).

 Note

Este procedimento mostra como adicionar um estágio de compilação entre os estágios Source (Origem) e Beta (Beta) ao pipeline. Para adicionar uma ação de compilação a um estágio existente, selecione Edit stage (Editar estágio) no estágio e avance para a

etapa 8 deste procedimento. Para adicionar o estágio de compilação em outro local, selecione Add stage (Adicionar estágio) no local desejado.



8. Para Stage name (Nome do estágio), insira o nome do estágio de compilação (por exemplo, **Build**). Se escolher um nome diferente, use-o nas etapas deste procedimento.
9. Em vez do estágio selecionado, selecione Add action (Adicionar ação).

Note

Este procedimento mostra como acrescentar a ação de compilação em um estágio de compilação. Para adicionar a ação de compilação em outro local, selecione Add action (Adicionar ação) no local desejado. Talvez seja necessário selecionar primeiro Edit stage (Editar estágio) no estágio existente em que deseja adicionar a ação de compilação.

10. Em Edit action (Editar ação), em Action name (Nome da ação), insira um nome para a ação (por exemplo, **CodeBuild**). Se escolher um nome diferente, use-o nas etapas deste procedimento.
11. Em Action provider (Provedor de ação), selecione CodeBuild.
12. Se você já tiver um projeto de compilação que deseja usar, em Nome do projeto, escolha o nome do projeto de compilação e avance para a próxima etapa deste procedimento.

Se você precisar criar um novo projeto de CodeBuild compilação, siga as instruções [Criar um projeto de compilação \(console\)](#) e retorne a esse procedimento.

Se você escolher um projeto de construção existente, ele deverá ter configurações de artefato de saída de construção já definidas (mesmo que as CodePipeline substitua). Para obter mais informações, consulte a descrição de Artifacts (Artefatos) em [Criar um projeto de compilação \(console\)](#) ou [Alterar configurações de um projeto de compilação \(console\)](#).

 Important

Se você habilitar webhooks para um CodeBuild projeto e o projeto for usado como uma etapa de compilação CodePipeline, duas compilações idênticas serão criadas para cada confirmação. Uma compilação é acionada por meio de webhooks e outra por meio do CodePipeline. Como o faturamento é por compilação, você é cobrado por ambas as compilações. Portanto, se você estiver usando CodePipeline, recomendamos que você desative os webhooks em CodeBuild. No CodeBuild console, desmarque a caixa do Webhook. Para ter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#).

13. Em Input artifacts (Artefatos de entrada), selecione o artefato de saída que você anotou anteriormente neste procedimento.
14. Em Output artifacts (Artefatos de saída), insira um nome para o artefato de saída (por exemplo, **MyAppBuild**).
15. Selecione Adicionar ação.
16. Escolha Save (Salvar) e selecione Save (Salvar) para salvar suas alterações no pipeline.
17. Selecione Release change.
18. Depois de o pipeline ser executado com sucesso, você pode obter o artefato de saída de build. Com o pipeline exibido no CodePipeline console, na ação Criar, escolha a dica de ferramenta. Anote o valor do artefato de saída (por exemplo, MyAppBuild).

 Note

Você também pode obter o artefato de saída da compilação escolhendo o link Construir artefatos na página de detalhes da compilação no CodeBuild console. Para acessar essa página, consulte [Visualizar detalhes de compilação \(console\)](#) e avance para a etapa 31 deste procedimento.

19. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

20. Na lista de bucket, abra o bucket usado pelo pipeline. O nome do bucket deve seguir o formato `codepipeline-region-ID-random-number`. Você pode usar o AWS CLI para executar o CodePipeline `get-pipeline` comando para obter o nome do bucket:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Na saída, o objeto `pipeline` contém um objeto `artifactStore`, que contém um valor `location` com o nome do bucket.

21. Abra a pasta que corresponda ao nome de seu pipeline (dependendo do tamanho do nome do pipeline, o nome da pasta pode estar truncado) e abra a pasta correspondente ao valor de `Output artifact` (Artefato de entrada) que você anotou anteriormente neste procedimento.
22. Extraia o conteúdo do arquivo. Se houver vários arquivos na pasta, extraia o conteúdo do arquivo com o carimbo Última modificação mais recente. (Talvez você precise atribuir ao arquivo a extensão `.zip`, de maneira que possa trabalhar com ele no utilitário ZIP do sistema.) O artefato de saída de compilação está no conteúdo extraído do arquivo.
23. Se você tiver instruído CodePipeline a implantar o artefato de saída de compilação, use as instruções do provedor de implantação para acessar o artefato de saída de compilação nos destinos de implantação.

Adicionar uma ação CodeBuild de teste a um pipeline (CodePipeline console)

1. Faça login no AWS Management Console usando:
 - Sua conta AWS root. Isso não é recomendado. Para obter mais informações, consulte [O usuário raiz da conta](#) no Guia do usuário do .
 - Um usuário administrador em sua AWS conta. Para obter mais informações, consulte [Como criar seu primeiro usuário e grupo Conta da AWS raiz](#) no Guia do usuário.
 - Um usuário em sua AWS conta com permissão para realizar o seguinte conjunto mínimo de ações:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
```

```
s3:GetObject  
s3:ListAllMyBuckets  
s3:ListBucket  
s3:PutBucketPolicy  
codecommit:ListBranches  
codecommit:ListRepositories  
codedeploy:GetApplication  
codedeploy:GetDeploymentGroup  
codedeploy:ListApplications  
codedeploy:ListDeploymentGroups  
elasticbeanstalk:DescribeApplications  
elasticbeanstalk:DescribeEnvironments  
lambda:GetFunctionConfiguration  
lambda:ListFunctions  
opsworks:DescribeStacks  
opsworks:DescribeApps  
opsworks:DescribeLayers
```

2. Abra o CodePipeline console em <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. No seletor de AWS região, escolha a AWS região em que seu funil está localizado. Essa deve ser uma AWS região CodeBuild com suporte. Para ter mais informações, consulte [AWS CodeBuild](#) no Referência geral da Amazon Web Services.
4. Na página Pipelines, selecione o nome do pipeline.
5. Na página de detalhes do pipeline, na ação Source (Origem), selecione a dica de ferramenta. Anote o valor do artefato de saída (por exemplo, MyApp).

 Note

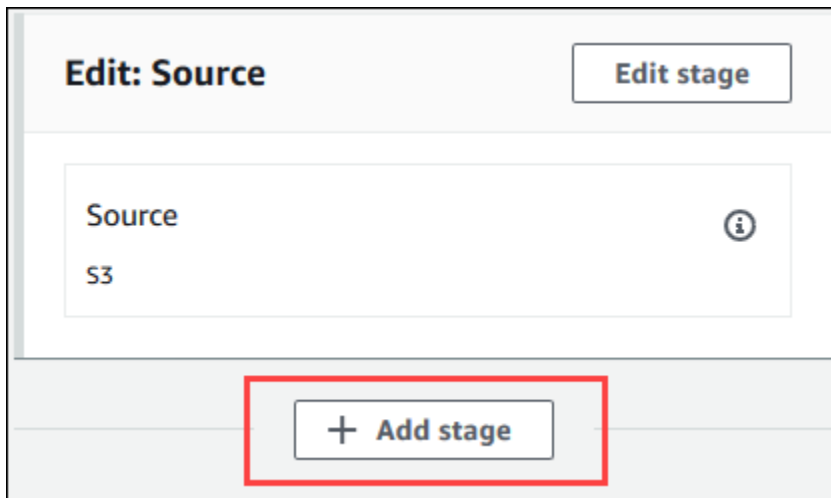
Este procedimento mostra como adicionar uma ação de teste em um estágio de teste entre os estágios Source (Origem) e Beta (Beta). Se desejar adicionar a ação de teste em outro local, aponte o mouse na ação próxima e anote o valor para Output artifact.

6. Selecione Editar.
7. Imediatamente depois do estágio Source (Origem), selecione Add stage (Adicionar estágio).

 Note

Este procedimento também mostra como adicionar um estágio de teste imediatamente após o estágio Source (Origem) do pipeline. Para adicionar uma ação de teste a um estágio existente, selecione Edit stage (Editar estágio) no estágio e avance para a etapa

8 deste procedimento. Para adicionar o estágio de teste em outro local, selecione Add stage (Adicionar estágio) no local desejado.



8. Em Stage name (Nome do estágio), insira o nome do estágio de teste (por exemplo, **Test**). Se escolher um nome diferente, use-o nas etapas deste procedimento.
9. No estágio selecionado, selecione + Add action (+ Adicionar ação).

Note

Este procedimento mostra como adicionar uma ação de teste em um estágio de teste. Para adicionar a ação de teste em outro local, selecione Add action (Adicionar ação) no local desejado. Você pode precisar selecionar primeiro Edit (Editar) no estágio existente onde deseja adicionar a ação de teste.

10. Em Edit action (Editar ação), em Action name (Nome da ação), insira um nome para a ação (por exemplo, **Test**). Se escolher um nome diferente, use-o nas etapas deste procedimento.
11. Em Action provider (Provedor de ação), em Test (Teste) escolha CodeBuild.
12. Se você já tiver um projeto de compilação que deseja usar, em Nome do projeto, escolha o nome do projeto de compilação e avance para a próxima etapa deste procedimento.

Se você precisar criar um novo projeto de CodeBuild compilação, siga as instruções [Criar um projeto de compilação \(console\)](#) e retorne a esse procedimento.

 Important

Se você habilitar webhooks para um CodeBuild projeto e o projeto for usado como uma etapa de compilação CodePipeline, duas compilações idênticas serão criadas para cada confirmação. Uma compilação é acionada por meio de webhooks e outra por meio do CodePipeline. Como o faturamento é por compilação, você é cobrado por ambas as compilações. Portanto, se você estiver usando CodePipeline, recomendamos que você desative os webhooks em. CodeBuild No CodeBuild console, desmarque a caixa do Webhook. Para ter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#)

13. Em Input artifacts (Artefatos de entrada), selecione o valor de Output artifact (Artefato de saída) que você anotou anteriormente neste procedimento.
14. (Opcional) Se quiser que a ação de teste produza um artefato de saída e você tiver configurado a buildspec adequadamente, em Output artifact (Artefato de saída), insira o valor que deseja atribuir ao artefato de saída.
15. Escolha Salvar.
16. Selecione Release change.
17. Depois de o pipeline ser executado com sucesso, você pode obter os resultados do teste. No estágio de teste do pipeline, escolha o CodeBuildhiperlink para abrir a página do projeto de construção relacionado no CodeBuild console.
18. Na página de projeto de compilação, na área Build history (Histórico da compilação), selecione o hyperlink Build run (Execução de compilação) relacionado.
19. Na página de execução da compilação, em Registros de compilação, escolha o hiperlink Exibir registro inteiro para abrir o registro de compilação no CloudWatch console da Amazon.
20. Observe os resultados do teste no log de build.

Use AWS CodeBuild com Codecov

O Codecov é uma ferramenta que mede a cobertura de teste do seu código. O Codecov identifica quais métodos e instruções em seu código não são testados. Use os resultados para determinar onde escrever testes para melhorar a qualidade do seu código. O Codecov está disponível para três dos repositórios de origem suportados por CodeBuild: GitHub, GitHub Enterprise Server e

Bitbucket. Se o seu projeto de compilação usa o GitHub Enterprise Server, você deve usar o Codecov Enterprise.

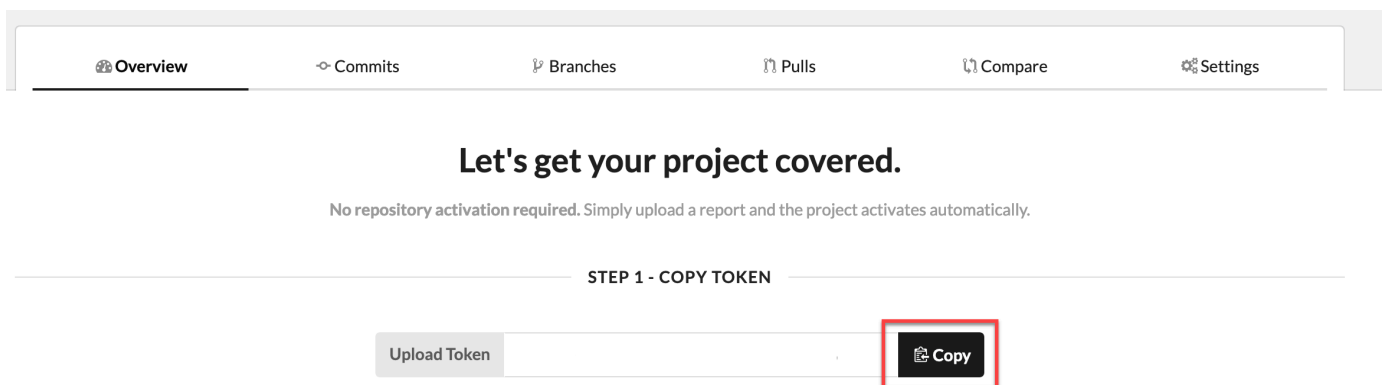
Quando você executa uma compilação de um CodeBuild projeto integrado ao Codecov, os relatórios do Codecov que analisam o código em seu repositório são enviados para o Codecov. Os logs de compilação incluem um link para os relatórios. Este exemplo mostra como integrar um projeto de compilação Python e um Java com o Codecov. Para obter uma lista de idiomas compatíveis com o Codecov, consulte [Idiomas compatíveis pelo Codecov](#) no site do Codecov.

Integrar o Codecov em um projeto de compilação

Use o procedimento a seguir para integrar o Codecov em um projeto de compilação.

Como integrar o Codecov com seu projeto de compilação

1. Acesse <https://codecov.io/signup> e cadastre-se em um GitHub repositório de origem do Bitbucket. Se você usa GitHub Enterprise, consulte [Codecov Enterprise](#) no site da Codecov.
2. No Codecov, adicione o repositório para o qual você deseja cobertura.
3. Quando as informações de token forem exibidas, escolha Copy (Copiar).



4. Adicione o token copiado como uma variável de ambiente chamada CODECOV_TOKEN ao seu projeto de compilação. Para obter mais informações, consulte [Alterar configurações de um projeto de compilação \(console\)](#).
5. Crie um arquivo de texto chamado `my_script.sh` em seu repositório. Insira o seguinte no arquivo:

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

- Escolha a guia Python ou Java conforme apropriado para o uso do seu projeto de compilação e siga estas etapas.

Java

- Adicione o seguinte JaCoCo pom.xml plug-in ao seu repositório.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.2</version>
      <executions>
        <execution>
          <goals>
            <goal>prepare-agent</goal>
          </goals>
        </execution>
        <execution>
          <id>report</id>
          <phase>test</phase>
          <goals>
            <goal>report</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

- Insira os seguintes comandos em seu arquivo buildspec. Para obter mais informações, consulte [Sintaxe de buildspec](#).

```
build:
  - mvn test -f pom.xml -fn
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

Python

Insira os seguintes comandos em seu arquivo buildspec. Para obter mais informações, consulte [Sintaxe de buildspec](#).

```
build:
  - pip install coverage
  - coverage run -m unittest discover
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

7. Execute uma compilação do seu projeto de compilação. Um link para relatórios do Codecov gerados para seu projeto aparece em seus logs de compilação. Use o link para visualizar os relatórios do Codecov. Para ter mais informações, consulte [Execute AWS CodeBuild compilações manualmente](#) e [Registre chamadas de AWS CodeBuild API com AWS CloudTrail](#). As informações do Codecov nos logs de compilação são semelhantes às seguintes:

```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh
```

```

  ____  _
 / ____| | | |
| |  ___| | | |
| | / __ \ | | |
| | \__ \ | | |
| | ___| | | |
|_| \___|_|_|_|

```

Bash-20200303-bc4d7e6

```
.[0;90m==>.[0m AWS CodeBuild detected.
```

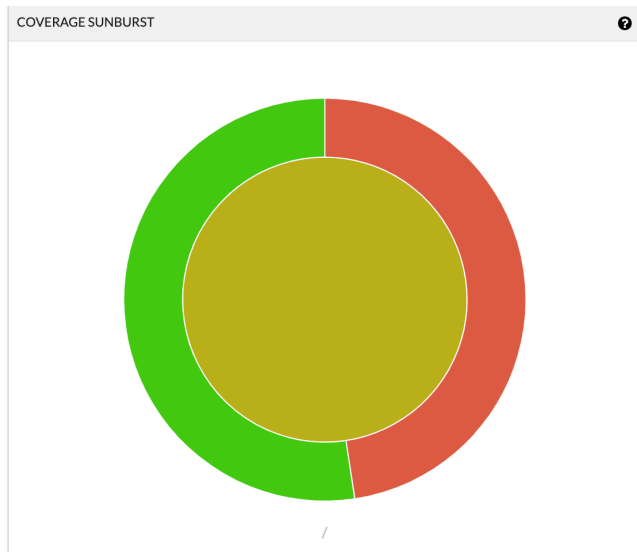
```
... The full list of Codecov log entries has been omitted for brevity ...
```

```
.
```

```
.[0;32m->.[0m View reports at .[0;36mhttps://codecov.io/github/user/test\_py/commit/commit-id.[0m
```

```
[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

Os relatórios são semelhantes aos seguintes:



Files					Coverage
<code>code.py</code>	10	7	0	3	70.00%
<code>tests.py</code>	11	11	0	0	100.00%
Project Totals (2 files)	21	18	0	3	85.71%

Use AWS CodeBuild com Jenkins

Você pode usar o plug-in Jenkins para AWS CodeBuild fazer a integração CodeBuild com seus trabalhos de construção do Jenkins. Em vez de você enviar seus trabalhos de build para os nós de build do Jenkins, o plugin envia os trabalhos para o CodeBuild. Isso elimina a necessidade de você provisionar, configurar e gerenciar nós de build do Jenkins.

Tópicos

- [Configurar o Jenkins](#)
- [Instalar o plug-in](#)
- [Usar o plug-in](#)

Configurar o Jenkins

Para obter informações sobre como configurar o Jenkins com o AWS CodeBuild plug-in e fazer o download do código-fonte do plug-in, consulte <https://github.com/aws-labs/aws-codebuild-jenkins-plugin>.

Instalar o plug-in

Se você já tem um servidor do Jenkins configurado e quer apenas instalar o plug-in do AWS CodeBuild, em sua instância do Jenkins, no Gerenciador de plug-ins, procure o **CodeBuild Plugin for Jenkins**.

Usar o plug-in

Para usar AWS CodeBuild com fontes de fora de uma VPC

1. Crie um projeto no CodeBuild console. Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#).
 - Escolha a AWS região em que você deseja executar a compilação.
 - (Opcional) Defina a configuração da Amazon VPC para permitir que o contêiner de CodeBuild compilação acesse recursos em sua VPC.
 - Anote o nome do projeto. Você vai precisar dele na etapa 3.
 - (Opcional) Se seu repositório de origem não for suportado nativamente pelo CodeBuild, você pode definir o Amazon S3 como o tipo de fonte de entrada para seu projeto.
2. No IAMconsole, crie um usuário para ser usado pelo plug-in Jenkins.
 - Ao criar credenciais para o usuário, selecione Programmatic Access (Acesso programático).
 - Crie uma política semelhante à seguinte e, em seguida, anexe-a ao usuário.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": ["arn:aws:logs:{{region}}:{{awsAccountId}}:log-group:/aws/codebuild/{{projectName}}:*"],
      "Action": ["logs:GetLogEvents"]
    },
    {
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::{{inputBucket}}"],
      "Action": ["s3:GetBucketVersioning"]
    },
    {
      "Effect": "Allow",
```

```
    "Resource": ["arn:aws:s3:::{{inputBucket}}/{{inputObject}}"],
    "Action": ["s3:PutObject"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::{{outputBucket}}/*"],
    "Action": ["s3:GetObject"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:codebuild:{{region}}:{{awsAccountId}}:project/{{projectName}}"],
    "Action": ["codebuild:StartBuild",
               "codebuild:BatchGetBuilds",
               "codebuild:BatchGetProjects"]
  }
]
```

3. Crie um projeto de estilo livre no Jenkins.

- Na página Configure (Configurar), selecione Add build step (Adicionar etapa de compilação), depois selecione Run build on CodeBuild (Executar compilação no &ACB;).
- Configure a etapa de compilação.
 - Informe os valores de Região, Credenciais e Nome do projeto.
 - Selecione Use Project source (Usar origem do projeto).
 - Salve a configuração e execute uma compilação a partir do Jenkins.

4. Em Source Code Management (Gerenciamento do código de origem), selecione como deseja obter a origem. Talvez seja necessário instalar o GitHub plug-in (ou o plug-in Jenkins para seu provedor de repositório de origem) no seu servidor Jenkins.

- Na página Configure (Configurar), selecione Add build step (Adicionar etapa de compilação), depois selecione Run build on AWS CodeBuild(Executar compilação no &ACB;).
- Configure a etapa de compilação.
 - Informe os valores de Região, Credenciais e Nome do projeto.
 - Selecione Use Jenkins source (Usar origem do Jenkins).
 - Salve a configuração e execute uma compilação a partir do Jenkins.

Para usar o AWS CodeBuild plug-in com o plug-in Jenkins pipeline

- Na página do projeto do pipeline do Jenkins, use o gerador de trechos para gerar um script de pipeline que é adicionado CodeBuild como uma etapa em seu pipeline. Ele deve gerar um script semelhante a este:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',  
sourceControlType: 'jenkins'
```

Use AWS CodeBuild com aplicativos sem servidor

O AWS Serverless Application Model (AWS SAM) é uma estrutura de código aberto para criar aplicativos sem servidor. Para obter mais informações, consulte o repositório do [modelo de aplicativo AWS sem servidor](#) em GitHub

Você pode usar AWS CodeBuild para empacotar e implantar aplicativos sem servidor que seguem o AWS SAM padrão. Para a etapa de implantação, CodeBuild pode usar AWS CloudFormation. Para automatizar a criação e a implantação de aplicativos sem servidor com CodeBuild e AWS CloudFormation, você pode usar AWS CodePipeline

Para obter mais informações, consulte [Deploying Serverless Applications](#), no Guia do desenvolvedor do AWS Serverless Application Model .

Recursos relacionados

- Para obter informações sobre como começar a usar AWS CodeBuild, consulte [Começando a AWS CodeBuild usar o console](#).
- Para obter informações sobre a solução de problemas em CodeBuild, consulte [Solução de problemas AWS CodeBuild](#).
- Para obter informações sobre cotas em CodeBuild, consulte [Cotas para AWS CodeBuild](#).

Avisos de terceiros AWS CodeBuild para Windows

Ao usar CodeBuild para compilações do Windows, você tem a opção de usar alguns pacotes e módulos de terceiros para permitir que seu aplicativo criado seja executado em sistemas operacionais Microsoft Windows e interopere com alguns produtos de terceiros. A lista a seguir

contém os termos legais aplicáveis de terceiros que controlam o uso de pacotes e módulos de terceiros especificados.

Tópicos

- [1\) Imagem do Docker de base: windowsservercore](#)
- [2\) Imagem do Docker baseada em Windows: choco](#)
- [3\) Imagem do Docker baseada em Windows: git – versão 2.16.2](#)
- [4\) imagem Docker baseada em Windows— --versão 15.0.26320.2 microsoft-build-tools](#)
- [5\) Imagem do Docker baseada em Windows: nuget.commandline – versão 4.5.1](#)
- [7\) Imagem do Docker baseada em Windows: netfx-4.6.2-devpack](#)
- [8\) Imagem do Docker baseada em Windows: visualfsharp tools, v 4.0](#)
- [9\) imagem Docker baseada em janelas — -4.6 netfx-pcl-reference-assemblies](#)
- [10\) Imagem do Docker baseada em Windows: visualcppbuildtools v 14.0.25420.1](#)
- [11\) Imagem do Docker baseada em Windows — 3-ondemand-package.cab microsoft-windows-netfx](#)
- [12\) Imagem do Docker baseada em Windows: dotnet-sdk](#)

1) Imagem do Docker de base: windowsservercore

(termos da licença disponíveis em: https://hub.docker.com/_/microsoft-windows-servercore)

Licença: ao solicitar e usar essa imagem de SO do contêiner para contêineres do Windows, você reconhece, compreende e consente com os seguintes Termos de Licença Adicionais:

TERMOS DE LICENÇA ADICIONAIS DO SOFTWARE MICROSOFT

IMAGEM DE SO DO CONTÊINER

A Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora), mencionada como "nós" ou "Microsoft", licencia esse suplemento de Imagem de SO do contêiner para você ("Suplementar"). Você está licenciado para usar este Suplemento em conjunto com o software do sistema operacional de host subjacente ("Software de Host") exclusivamente para ajudar na execução do recurso de contêineres do Software de Host. Os termos de licença de Software de Host aplicam-se ao uso do Suplemento. Não é possível usá-lo se você não tiver uma licença para o Software de Host. Você pode usar este Suplemento com cada cópia licenciada válida do Software de Host.

REQUISITOS ADICIONAIS DE LICENCIAMENTO E/OU DIREITOS DE USO

O uso do Suplemento, conforme especificado no parágrafo anterior pode resultar na criação ou na modificação de uma imagem de contêiner ("Imagem de Contêiner"), que inclui alguns componentes do Suplemento. Para maior clareza, uma Imagem de Contêiner é separada e distinta de uma máquina virtual ou imagem do dispositivo virtual. De acordo com estes termos de licença, concedemos a você um direito restrito para redistribuir esses componentes do Suplemento sob as seguintes condições:

- (i) você pode usar os componentes do Suplemento somente quando usado em e como parte de sua Imagem de Contêiner;
- (ii) você pode usar esses componentes do Suplemento em sua Imagem de Contêiner contanto que tenha funcionalidades primárias significativas em sua Imagem de Contêiner materialmente separadas e distintas do Suplemento; e
- (iii) você concorda em incluir esses termos de licença (ou termos semelhantes exigidos por nós ou por um hoster) em sua Imagem de Contêiner para licenciar corretamente o possível uso dos componentes do Suplemento por seus usuários finais.

Reservamos todos os outros direitos não expressamente concedidos neste documento.

Ao usar este Suplemento, você aceita estes termos. Se você não aceitá-los, não use este Suplemento.

[Como parte dos Termos de Licença Suplementar para esses contêineres Container OS Image para Windows, você também está sujeito aos termos subjacentes da licença de software host do Windows Server, que estão localizados em: <https://www.microsoft.com/en-us/usetterms>.](https://www.microsoft.com/en-us/usetterms)

2) Imagem do Docker baseada em Windows: choco

(termos da licença disponíveis em: <https://github.com/chocolatey/choco/blob/master/LICENSE>)

Direitos autorais 2011 - Present RealDimensions Software, LLC

Licenciado segundo a Licença do Apache, versão 2.0 (a "Licença"). Não é possível usar estes arquivos, exceto em conformidade com a Licença. Você poderá obter uma cópia da Licença em

<http://www.apache.org/licenses/LICENSE-2,0>

A menos que exigido por lei aplicável ou acordado por escrito, o software distribuído sob a Licença é distribuído no "ESTADO EM QUE SE ENCONTRA", SEM GARANTIAS OU CONDIÇÕES DE

QUALQUER TIPO, sejam expressas ou implícitas. Consulte a licença para conhecer as permissões e limitações específicas do idioma na licença.

3) Imagem do Docker baseada em Windows: git – versão 2.16.2

(termos da licença disponíveis em: <https://chocolatey.org/packages/git/2.16.2>)

Licenciado sob a Licença Pública Geral GNU, versão 2, disponível em: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

4) imagem Docker baseada em Windows— --versão 15.0.26320.2 microsoft-build-tools

(termos da licença disponíveis em: <https://www.visualstudio.com/license-terms/mt171552/>)

EXTENSÕES DO MICROSOFT VISUAL STUDIO 2015, SHELLS E REDISTRIBUÍVEIS DO VISUAL STUDIO C++

Esses termos de licença são um acordo entre a Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora) e você. Eles se aplicam ao software mencionado acima. Os termos também se aplicam a quaisquer serviços ou atualizações da Microsoft para o software, exceto quando tenham termos adicionais.

SE ESTIVER EM CONFORMIDADE COM ESTES TERMOS DE LICENÇA, VOCÊ POSSUI OS DIREITOS ABAIXO.

1. DIREITOS DE INSTALAÇÃO E DE USO. Você pode instalar e usar qualquer número de cópias do software.
2. TERMOS PARA COMPONENTES ESPECÍFICOS.
 - a. Utilitários. O software pode conter alguns itens na Lista de utilitários em <https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs>. Você pode copiar e instalar esses itens, se incluídos no software, em suas máquinas ou em outras máquinas de terceiros, para depurar e implantar seus aplicativos e bancos de dados que você desenvolveu com o software. Observe que os Utilitários são criados para uso temporário, a Microsoft pode não conseguir aplicar patches ou atualizar os Utilitários separadamente do restante do software e alguns Utilitários, por natureza, podem permitir que outros acessem máquinas nas quais

estão instalados. Como resultado, é necessário excluir todos os utilitários instalados após concluir a depuração ou a implantação de seus aplicativos e bancos de dados. A Microsoft não é responsável pelo uso de terceiros ou pelo acesso de utilitários instalados em qualquer máquina.

- b. Plataformas Microsoft. O software pode incluir componentes do Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office e Microsoft SharePoint. Esses componentes são regidos por contratos separados e suas próprias políticas de suporte ao produto, conforme descrito nos termos de licença encontrados no diretório de instalação desse componente ou na pasta "Licenças" que acompanha o software.
- c. Componentes de terceiros. O software pode incluir componentes de terceiros com avisos legais separados ou regidos por outros contratos, conforme descrito no ThirdPartyNotices arquivo que acompanha o software. Mesmo que tais componentes sejam regidos por outros acordos, as isenções de responsabilidade e as limitações e exclusões de danos abaixo também se aplicam. O software também pode incluir componentes licenciados sob licenças de código aberto com obrigações de disponibilidade de código-fonte. Cópias dessas licenças, se aplicável, estão incluídas no ThirdPartyNotices arquivo. Podemos fornecer esse código-fonte a você, se e conforme exigido pelas licenças de código aberto relevantes, você enviar uma ordem de pagamento ou cheque de US\$ 5,00 para: Equipe de Conformidade do Código-Fonte, Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Escreva o código-fonte para um ou mais dos componentes listados abaixo na linha de memorando do seu pagamento:
 - Ferramentas remotas para Visual Studio 2015;
 - Profiler individual para Visual Studio 2015;
 - IntelliTraceCollector para Visual Studio 2015;
 - Microsoft VC++ Redistributable 2015;
 - Biblioteca MFC multibyte para Visual Studio 2015;
 - Microsoft Build Tools 2015;
 - Feedback Client;
 - Shell do Visual Studio 2015 integrado; ou
 - Shell do Visual Studio 2015 isolado.

Também podemos fazer uma cópia do código-fonte disponível em <http://thirdpartysource.microsoft.com>.

- 3. DADOS. O software pode coletar informações sobre você e seu uso do software e enviá-las para a Microsoft. A Microsoft pode usar essas informações para fornecer serviços e aprimorar

nossos produtos e serviços. Você pode excluir muitos desses cenários, mas não todos, conforme descrito na documentação do produto. Há também alguns recursos no software que podem permitir que você colete dados de usuários de seus aplicativos. Se você usar esses recursos para ativar a coleta de dados em seus aplicativos, é necessário estar em conformidade com a lei aplicável, incluindo o fornecimento de avisos apropriados aos usuários de seus aplicativos. Você pode aprender mais sobre coleta e uso de dados na documentação de ajuda e na declaração de privacidade em <https://privacy.microsoft.com/en-us/privacystatement>. O uso do software funciona como seu consentimento para essas práticas.

4. **ESCOPO DA LICENÇA.** O software é licenciado, não vendido. Este contrato só lhe concede alguns direitos para usar o software. A Microsoft reserva todos os outros direitos. A menos que a lei aplicável lhe conceda mais direitos, apesar desta limitação, você poderá usar o software somente conforme expressamente permitido neste contrato. Ao fazê-lo, é necessário obedecer a quaisquer limitações técnicas do software que permitam usá-lo apenas de determinadas maneiras. Não é permitido
 - contornar as limitações técnicas do software;
 - reverter a engenharia, descompilar ou desmontar o software ou tentar fazer isso, exceto e somente na medida exigida por termos de licenciamento de terceiros que regem o uso de certos componentes de código aberto que podem estar incluídos no software;
 - remover, minimizar, bloquear ou modificar quaisquer avisos da Microsoft ou de seus fornecedores no software;
 - usar o software de alguma maneira que seja contra a lei; ou
 - compartilhar, publicar, alugar ou arrendar o software ou fornecer o software como uma solução autônoma hospedada como solução para outras pessoas usarem.
5. **RESTRIÇÕES DE EXPORTAÇÃO.** É necessário cumprir todas as leis e regulamentações de exportação nacionais e internacionais que se aplicam ao software, que incluem restrições sobre destinos, usuários finais e uso final. Para obter mais informações sobre restrições à exportação, visite (aka.ms/exporting).
6. **SERVIÇOS DE SUPORTE.** Como esse software é fornecido "no estado em que se encontra", não podemos fornecer serviços de suporte para ele.
7. **ACORDO INTEGRAL.** Este contrato e os termos para suplementos, atualizações, serviços baseados na Internet e serviços de suporte que você usa são o contrato integral para os serviços de software e suporte.
8. **LEI APLICÁVEL.** Se você adquiriu o software nos Estados Unidos, a lei de Washington se aplica à interpretação e às reclamações de violação deste contrato, e as leis do estado onde você reside

se aplicam a todas as outras reivindicações. Se você adquiriu o software em qualquer outro país, suas leis se aplicam.

9. DIREITOS DO CONSUMIDOR; VARIAÇÕES REGIONAIS. Este acordo descreve determinados direitos legais. Você pode ter outros direitos, incluindo direitos do consumidor, de acordo com as leis do seu estado ou país. Além do seu relacionamento com a Microsoft, você também pode ter direitos com relação à parte da qual adquiriu o software. Este contrato não altera esses outros direitos se as leis do seu estado ou país não permitirem que isso aconteça. Por exemplo, se você adquiriu o software em uma das regiões abaixo, ou se a lei obrigatória do país se aplica, as seguintes provisões se aplicam a você:

- a. Austrália. Você tem garantias estatutárias de acordo com a Lei do Consumidor Australiana e nada neste contrato tem a intenção de afetar esses direitos.
- b. Canadá. Se você adquiriu este software no Canadá, pode parar de receber atualizações desativando o recurso de atualização automática, desconectando seu dispositivo da Internet (se e quando você se reconectar à Internet, no entanto, o software continuará a procurar e instalar atualizações) ou desinstalando o software. A documentação do produto, se houver, também pode especificar como desativar as atualizações para seu dispositivo ou software específico.
- c. Alemanha e Áustria.
 - i. Garantia. O software devidamente licenciado será executado substancialmente conforme descrito em todos os materiais da Microsoft que acompanham o software. No entanto, a Microsoft não oferece garantia contratual em relação ao software licenciado.
 - ii. Limitação de responsabilidade. Em caso de conduta intencional, negligência grave, reclamações baseadas na Lei de Responsabilidade pelo Produto, bem como, em caso de morte ou lesão corporal, a Microsoft é responsável de acordo com a lei estatutária. Sujeito à cláusula anterior (ii), a Microsoft só será responsável por negligência leve se ela violar tais obrigações contratuais relevantes, cujo cumprimento facilitará a devida execução deste contrato, cuja violação colocaria em risco a finalidade deste contrato e a conformidade com a qual uma parte pode confiar constantemente (as chamadas "obrigações cardinais"). Em outros casos de negligência leve, a Microsoft não será responsável.

10)SENÇÃO DE GARANTIA. O SOFTWARE É LICENCIADO NO "ESTADO EM QUE SE ENCONTRA". VOCÊ ARCA COM O RISCO DE USÁ-LO. A MICROSOFT NÃO OFERECE GARANTIAS OU CONDIÇÕES EXPRESSAS. DESDE QUE PERMITIDO PELAS LEIS LOCAIS, A MICROSOFT EXCLUI AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA E NÃO VIOLAÇÃO.

11 LIMITAÇÃO E EXCLUSÃO DE DANOS. VOCÊ PODE RECUPERAR DA MICROSOFT E SEUS FORNECEDORES SOMENTE DANOS DIRETOS ATÉ US\$ 5,00. É VEDADO PLEITEAR INDENIZAÇÃO POR QUAISQUER OUTROS DANOS, INCLUINDO LUCROS CESSANTES, DANOS CONSEQUENTES, ESPECIAIS, INDIRETOS OU INCIDENTAIS. Essa limitação se aplica a (a) qualquer assunto relacionado ao software, serviços, conteúdo (incluindo código) em sites da Internet de terceiros ou aplicativos de terceiros; e (b) reclamações por violação de contrato, violação de garantia ou condição, responsabilidade estrita, negligência ou outro ato ilícito na medida permitida pela lei aplicável.

Também se aplica mesmo que a Microsoft saiba ou deva saber sobre a possibilidade dos danos. A limitação ou exclusão acima pode não se aplicar a você porque seu país pode não permitir a exclusão ou limitação de danos incidentais, consequentes ou outros.

ID do EULA: VS2 ShellsRedist 015_Update3_ _ <ENU>

5) Imagem do Docker baseada em Windows: nuget.commandline – versão 4.5.1

(termos da licença disponíveis em: <https://github.com/NuGet/Home/blob/dev/LICENSE.txt>)

Copyright (c) .NET Foundation. Todos os direitos reservados.

Licenciado segundo a Licença do Apache, versão 2.0 (a "Licença"). Não é possível usar estes arquivos, exceto em conformidade com a Licença. Você poderá obter uma cópia da Licença em

<http://www.apache.org/licenses/LICENSE-2,0>

A menos que exigido por lei aplicável ou acordado por escrito, o software distribuído sob a Licença é distribuído no "ESTADO EM QUE SE ENCONTRA", SEM GARANTIAS OU CONDIÇÕES DE QUALQUER TIPO, sejam expressas ou implícitas. Consulte a licença para conhecer as permissões e limitações específicas do idioma na licença.

7) Imagem do Docker baseada em Windows: netfx-4.6.2-devpack

TERMOS DE LICENÇA ADICIONAIS DO SOFTWARE MICROSOFT

.NET FRAMEWORK E PACOTES DE IDIOMA ASSOCIADOS PARA O SISTEMA OPERACIONAL DO MICROSOFT WINDOWS

A Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora) licencia este suplemento para você. Se você estiver licenciado para usar o software do sistema operacional Microsoft Windows (o "software"), poderá usar este suplemento. Não é possível usá-lo se você não tiver uma licença para o software. Você pode usar este suplemento com cada cópia licenciada válida do software.

Os seguintes termos de licença descrevem termos de uso adicionais para este suplemento. Estes termos e os termos de licença do software se aplicam ao uso do suplemento. Se houver um conflito, esses termos de licença adicionais serão aplicados.

AO USAR ESTE SUPLEMENTO, VOCÊ ACEITA ESTES TERMOS. SE VOCÊ NÃO OS ACEITAR, NÃO USE ESTE SUPLEMENTO.

Se estiver em conformidade com estes termos de licença, você possui os direitos abaixo.

1. CÓDIGO DISTRIBUÍVEL. O suplemento é composto por código distribuível. "Código Distribuível" é o código que você pode distribuir nos programas que desenvolve se cumprir os termos abaixo.
 - a. Direito de uso e distribuição.
 - Você pode copiar e distribuir o código objeto do suplemento.
 - Distribuição de terceiros. Você pode permitir que os distribuidores de seus programas copiem e distribuam o Código Distribuível como parte desses programas.
 - b. Requisitos de distribuição. Para qualquer código distribuível distribuído, você deve
 - adicionar funcionalidades primárias significativas aos seus programas;
 - para qualquer Código Distribuível que tenha uma extensão de nome de arquivo .lib, distribuir apenas os resultados da execução desse Código Distribuível através de um vinculador com seu programa;
 - distribuir o Código Distribuível incluído em um programa de configuração somente como parte desse programa sem modificação;
 - exigir que os distribuidores e usuários finais externos concordem com termos que o protegem tanto quanto este contrato;
 - exibir o aviso de direitos autorais válido em seus programas; e
 - indenizar, defender e isentar a Microsoft de quaisquer reclamações, incluindo honorários advocatícios, relacionadas à distribuição ou ao uso de seus programas.

c. Restrições de distribuição. Não é permitido

- alterar qualquer aviso de direitos autorais, marca registrada ou patente no Código Distribuível;
- usar as marcas registradas da Microsoft nos nomes de seus programas ou de uma forma que sugira que seus programas venham ou sejam endossados pela Microsoft;
- distribuir o Código Distribuível para ser executado em uma plataforma diferente da plataforma Windows;
- incluir o Código Distribuível em programas maliciosos, enganosos ou ilegais; ou
- modificar ou distribuir o código-fonte de qualquer Código Distribuível para que qualquer parte dele fique sujeita a uma Licença Excluída. Uma Licença Excluída é aquela que exige, como condição de uso, modificação ou distribuição, que
 - o código seja divulgado ou distribuído na forma de código-fonte; ou
 - outros tenham o direito de modificá-lo.

2. SERVIÇOS DE SUPORTE PARA SUPLEMENTO. A Microsoft fornece serviços de suporte para esse software conforme descrito em www.support.microsoft.com/common/international.aspx.

8) Imagem do Docker baseada em Windows: visualfsharpools, v 4.0

(termos da licença disponíveis em: <https://github.com/dotnet/fsharp/blob/main/License.txt>)

Copyright (c) Microsoft Corporation. Todos os direitos reservados.

Licenciado segundo a Licença do Apache, versão 2.0 (a "Licença"). Não é possível usar estes arquivos, exceto em conformidade com a Licença. Você poderá obter uma cópia da Licença em

<http://www.apache.org/licenses/LICENSE-2.0>

A menos que exigido por lei aplicável ou acordado por escrito, o software distribuído sob a Licença é distribuído no "ESTADO EM QUE SE ENCONTRA", SEM GARANTIAS OU CONDIÇÕES DE QUALQUER TIPO, sejam expressas ou implícitas. Consulte a licença para conhecer as permissões e limitações específicas do idioma na licença.

9) imagem Docker baseada em janelas — -4.6 netfx-pcl-reference-assemblies

TERMOS DE LICENÇA DO SOFTWARE MICROSOFT

CONJUNTOS DE REFERÊNCIA DA BIBLIOTECA DE CLASSE PORTÁTIL DO MICROSOFT .NET – 4.6

Esses termos de licença são um acordo entre a Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora) e você. Leia-os. Eles se aplicam ao software mencionado acima. Os termos também se aplicam a todos os itens da Microsoft a seguir:

- atualizações,
- suplementos,
- serviços baseados na Internet e
- serviços de suporte

para este software, a menos que outros termos acompanhem esses itens. Se esse for o caso, esses termos se aplicam.

AO USAR O SOFTWARE, VOCÊ ACEITA ESTES TERMOS. SE VOCÊ NÃO OS ACEITAR, NÃO USE O SOFTWARE.

SE ESTIVER EM CONFORMIDADE COM ESTES TERMOS DE LICENÇA, VOCÊ POSSUI OS DIREITOS VITALÍCIOS ABAIXO.

1. DIREITOS DE INSTALAÇÃO E DE USO. Você pode instalar e usar qualquer número de cópias do software para projetar, desenvolver e testar seus programas.
2. REQUISITOS ADICIONAIS DE LICENCIAMENTO E/OU DIREITOS DE USO
 - a. Código distribuível. Você pode distribuir o software em programas de ferramentas de desenvolvedor que desenvolver, para permitir que os clientes de seus programas desenvolvam bibliotecas portáteis para uso com qualquer dispositivo ou sistema operacional, se você cumprir os termos abaixo.
 - i. Direito de uso e distribuição. O software é “código distribuível”.
 - Código distribuível. Você pode copiar e distribuir o código objeto do software.
 - Distribuição de terceiros. Você pode permitir que os distribuidores de seus programas copiem e distribuam o Código Distribuível como parte desses programas.
 - ii. Requisitos de distribuição. Para qualquer código distribuível distribuído, você deve

- adicionar funcionalidades primárias significativas aos seus programas;
- exigir que os distribuidores e seus clientes concordem com termos que o protegem tanto quanto este contrato;
- exibir o aviso de direitos autorais válido em seus programas; e
- indenizar, defender e isentar a Microsoft de quaisquer reclamações, incluindo honorários advocatícios, relacionadas à distribuição ou ao uso de seus programas.

iii. Restrições de distribuição. Não é permitido

- alterar qualquer aviso de direitos autorais, marca registrada ou patente no Código Distribuível;
- usar as marcas registradas da Microsoft nos nomes de seus programas ou de uma forma que sugira que seus programas venham ou sejam endossados pela Microsoft;
- incluir o Código Distribuível em programas maliciosos, enganosos ou ilegais; ou
- modificar ou distribuir o Código Distribuível de modo que qualquer parte dele fique sujeita a uma Licença Excluída. Uma Licença Excluída é aquela que exige, como condição de uso, modificação ou distribuição, que
 - o código seja divulgado ou distribuído na forma de código-fonte; ou
 - outros tenham o direito de modificá-lo.

3. ESCOPO DA LICENÇA. O software é licenciado, não vendido. Este contrato só lhe concede alguns direitos para usar o software. A Microsoft reserva todos os outros direitos. A menos que a lei aplicável lhe conceda mais direitos, apesar desta limitação, você poderá usar o software somente conforme expressamente permitido neste contrato. Ao fazê-lo, é necessário obedecer a quaisquer limitações técnicas do software que permitam usá-lo apenas de determinadas maneiras. Não é permitido

- contornar as limitações técnicas do software;
- reverter a engenharia, descompilar ou desmontar o software, exceto e somente na medida em que a lei aplicável permitir expressamente, apesar dessa limitação;
- publicar o software para outros copiarem; ou
- alugar, arrendar ou emprestar o software.

4. FEEDBACK. Você pode fornecer feedback sobre o software. Se você fornecer feedback sobre o software à Microsoft, você concede à Microsoft, sem custos, o direito de usar, compartilhar e comercializar seu feedback de qualquer maneira e para qualquer finalidade. Você também concede a terceiros, sem custos, quaisquer direitos de patentes necessários para seus produtos, tecnologias e serviços para usar ou interagir com qualquer parte específica de um software

ou serviço da Microsoft que inclua o feedback. Você não fornecerá feedback sujeito a uma licença que exija que a Microsoft licencie seu software ou sua documentação para terceiros, pois incluiremos seu feedback neles. Estes direitos continuam em vigor independentemente deste acordo.

5. **TRANSFERÊNCIA DE TERCEIROS.** O primeiro usuário do software pode transferi-lo, além deste contrato, diretamente para um terceiro. Antes da transferência, essa parte deve concordar que este contrato se aplica à transferência e ao uso do software. O primeiro usuário deve desinstalar o software antes de transferi-lo separadamente do dispositivo. O primeiro usuário não pode reter nenhuma cópia.
6. **RESTRIÇÕES DE EXPORTAÇÃO.** O software está sujeito às leis e regulamentos de exportação dos Estados Unidos. Você deve estar em conformidade com todas as leis e regulamentos de exportação nacionais e internacionais que se aplicam ao software. Essas leis incluem restrições sobre destinos, usuários finais e uso final. Para obter informações adicionais, consulte www.microsoft.com/exporting.
7. **SERVIÇOS DE SUPORTE.** Como esse software é fornecido "no estado em que se encontra", não podemos fornecer serviços de suporte para ele.
8. **ACORDO INTEGRAL.** Este contrato e os termos para suplementos, atualizações, serviços baseados na Internet e serviços de suporte que você usa são o contrato integral para todos os serviços de software e suporte que você fornece.
9. **LEI APLICÁVEL.**
 - a. Estados Unidos. Se você adquiriu o software nos Estados Unidos, a legislação do estado de Washington rege a interpretação deste contrato e se aplica a reclamações por violação, independentemente dos princípios de conflito de leis. As leis do estado onde você mora regem todas as outras reivindicações, incluindo reivindicações de acordo com as leis estaduais de proteção ao consumidor, leis de concorrência desleal e ações ilícitas.
 - b. Fora dos Estados Unidos. Se você adquiriu o software em qualquer outro país, as leis desse país se aplicam.
10. **EFEITO JURÍDICO.** Este acordo descreve determinados direitos legais. Você pode ter outros direitos de acordo com as leis do seu país. Você também pode ter direitos com relação à parte de quem você adquiriu o software. Este contrato não altera seus direitos sob as leis de seu país se elas não permitirem isso.
11. **ISENÇÃO DE GARANTIA.** O SOFTWARE É LICENCIADO NO "ESTADO EM QUE SE ENCONTRA". VOCÊ ARCA COM O RISCO DE USÁ-LO. A MICROSOFT NÃO OFERECE GARANTIAS OU CONDIÇÕES EXPRESSAS. VOCÊ PODE TER DIREITOS ADICIONAIS DE CONSUMIDOR OU GARANTIAS ESTATUTÁRIAS REGIDOS PELAS LEIS LOCAIS QUE NÃO

PODEM SER ALTERADOS POR ESTE ACORDO. DESDE QUE PERMITIDO PELAS LEIS LOCAIS, A MICROSOFT EXCLUI AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA E NÃO VIOLAÇÃO.

PARA A AUSTRÁLIA – VOCÊ TEM GARANTIAS ESTATUTÁRIAS SOB A LEI AUSTRALIANA DO CONSUMIDOR E NADA NESTES TERMOS PRETENDE AFETAR ESSES DIREITOS.

12LIMITAÇÃO E EXCLUSÃO DE RECURSOS E DANOS. VOCÊ PODE RECUPERAR DA MICROSOFT E SEUS FORNECEDORES SOMENTE DANOS DIRETOS ATÉ US\$ 5,00. É VEDADO PLEITEAR INDENIZAÇÃO POR QUAISQUER OUTROS DANOS, INCLUINDO LUCROS CESSANTES, DANOS CONSEQUENTES, ESPECIAIS, INDIRETOS OU INCIDENTAIS.

Esta limitação aplica-se a

- tudo que estiver relacionado ao software, serviços, conteúdo (incluindo código) em sites da Internet de terceiros ou programas de terceiros;
- reclamações por violação de contrato, violação de garantia ou condição, responsabilidade estrita, negligência ou outro ato ilícito na medida permitida pela lei aplicável.

Também se aplica mesmo que a Microsoft saiba ou deva saber sobre a possibilidade dos danos. A limitação ou exclusão acima pode não se aplicar a você porque seu país pode não permitir a exclusão ou limitação de danos incidentais, consequentes ou outros.

10) Imagem do Docker baseada em Windows: visualcppbuildtools v 14.0.25420.1

(termos da licença disponíveis em: <https://www.visualstudio.com/license-terms/mt644918/>)

FERRAMENTAS DE COMPILAÇÃO DO MICROSOFT VISUAL C++

TERMOS DE LICENÇA DO SOFTWARE MICROSOFT

FERRAMENTAS DE COMPILAÇÃO DO MICROSOFT VISUAL C++

Esses termos de licença são um acordo entre a Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora) e você. Eles se aplicam ao software mencionado acima. Os termos também se aplicam a quaisquer serviços ou atualizações da Microsoft para o software, exceto na medida em que eles tenham termos diferentes.

SE ESTIVER EM CONFORMIDADE COM ESTES TERMOS DE LICENÇA, VOCÊ POSSUI OS DIREITOS ABAIXO.

1. DIREITOS DE INSTALAÇÃO E DE USO.

a. Um usuário pode usar cópias do software para desenvolver e testar seus aplicativos.

2. DADOS. O software pode coletar informações sobre você e seu uso do software e enviá-las para a Microsoft. A Microsoft pode usar essas informações para fornecer serviços e aprimorar nossos produtos e serviços. Você pode excluir muitos desses cenários, mas não todos, conforme descrito na documentação do produto. Há também alguns recursos no software que podem permitir que você colete dados de usuários de seus aplicativos. Se você usar esses recursos para ativar a coleta de dados em seus aplicativos, é necessário estar em conformidade com a lei aplicável, incluindo o fornecimento de avisos apropriados aos usuários de seus aplicativos. Você pode saber mais sobre a coleta e o uso de dados na documentação de ajuda e na declaração de privacidade em <http://go.microsoft.com/fwlink/?LinkID=528096>. O uso do software funciona como seu consentimento para essas práticas.

3. TERMOS PARA COMPONENTES ESPECÍFICOS.

- a. Servidor de compilação. O software pode conter alguns componentes do Build Server listados em BuildServer arquivos.TXT e/ou quaisquer arquivos listados na BuildeServer lista localizada de acordo com estes Termos de Licença de Software da Microsoft. Você pode copiar e instalar esses itens, se incluídos no software, em suas máquinas de compilação. Você e outras pessoas em sua organização podem usar esses itens em suas máquinas de compilação apenas com o propósito de compilar, criar, verificar e arquivar seus aplicativos ou executar testes de qualidade ou desempenho como parte do processo de criação.
- b. Plataformas Microsoft. O software pode incluir componentes do Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office e Microsoft SharePoint. Esses componentes são regidos por contratos separados e suas próprias políticas de suporte ao produto, conforme descrito nos termos de licença encontrados no diretório de instalação desse componente ou na pasta "Licenças" que acompanha o software.
- c. Componentes de terceiros. O software pode incluir componentes de terceiros com avisos legais separados ou regidos por outros contratos, conforme descrito no ThirdPartyNotices arquivo que acompanha o software. Mesmo que tais componentes sejam regidos por outros acordos, as isenções de responsabilidade e as limitações e exclusões de danos abaixo também se aplicam.
- d. Gerenciadores de pacotes. O software pode incluir gerenciadores de pacotes, como o Nuget, que oferecem a opção de fazer download de outros pacotes de software da Microsoft e de

terceiros para usar com o aplicativo. Esses pacotes estão sob suas próprias licenças e não sob este contrato. A Microsoft não distribui, licencia ou fornece quaisquer garantias para qualquer um dos pacotes de terceiros.

4. **ESCOPO DA LICENÇA.** O software é licenciado, não vendido. Este contrato só lhe concede alguns direitos para usar o software. A Microsoft reserva todos os outros direitos. A menos que a lei aplicável lhe conceda mais direitos, apesar desta limitação, você poderá usar o software somente conforme expressamente permitido neste contrato. Ao fazê-lo, é necessário obedecer a quaisquer limitações técnicas do software que permitam usá-lo apenas de determinadas maneiras. Para obter mais informações, consulte [https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms #1](https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms#1) -. installation-and-use-rights Não é permitido
 - contornar as limitações técnicas do software;
 - reverter a engenharia, descompilar ou desmontar o software ou tentar fazer isso, exceto e somente na medida exigida por termos de licenciamento de terceiros que regem o uso de certos componentes de código aberto que podem estar incluídos no software;
 - remover, minimizar, bloquear ou modificar quaisquer avisos da Microsoft ou de seus fornecedores;
 - usar o software de alguma maneira que seja contra a lei; ou
 - compartilhar, publicar, alugar ou arrendar o software ou fornecer o software como uma solução autônoma hospedada como solução para outras pessoas usarem.
5. **RESTRIÇÕES DE EXPORTAÇÃO.** É necessário cumprir todas as leis e regulamentações de exportação nacionais e internacionais que se aplicam ao software, que incluem restrições sobre destinos, usuários finais e uso final. Para obter mais informações sobre restrições à exportação, visite (aka.ms/exporting).
6. **SERVIÇOS DE SUPORTE.** Como esse software é fornecido "no estado em que se encontra", não podemos fornecer serviços de suporte para ele.
7. **ACORDO INTEGRAL.** Este contrato e os termos para suplementos, atualizações, serviços baseados na Internet e serviços de suporte que você usa são o contrato integral para os serviços de software e suporte.
8. **LEI APLICÁVEL.** Se você adquiriu o software nos Estados Unidos, a lei de Washington se aplica à interpretação e às reclamações de violação deste contrato, e as leis do estado onde você reside se aplicam a todas as outras reivindicações. Se você adquiriu o software em qualquer outro país, suas leis se aplicam.
9. **DIREITOS DO CONSUMIDOR; VARIAÇÕES REGIONAIS.** Este acordo descreve determinados direitos legais. Você pode ter outros direitos, incluindo direitos do consumidor, de acordo com as

leis do seu estado ou país. Além do seu relacionamento com a Microsoft, você também pode ter direitos com relação à parte da qual adquiriu o software. Este contrato não altera esses outros direitos se as leis do seu estado ou país não permitirem que isso aconteça. Por exemplo, se você adquiriu o software em uma das regiões abaixo, ou se a lei obrigatória do país se aplica, as seguintes provisões se aplicam a você:

- Austrália. Você tem garantias estatutárias de acordo com a Lei do Consumidor Australiana e nada neste contrato tem a intenção de afetar esses direitos.
- Canadá. Se você adquiriu este software no Canadá, pode parar de receber atualizações desativando o recurso de atualização automática, desconectando seu dispositivo da Internet (se e quando você se reconectar à Internet, no entanto, o software continuará a procurar e instalar atualizações) ou desinstalando o software. A documentação do produto, se houver, também pode especificar como desativar as atualizações para seu dispositivo ou software específico.
- Alemanha e Áustria.
 - Garantia. O software devidamente licenciado será executado substancialmente conforme descrito em todos os materiais da Microsoft que acompanham o software. No entanto, a Microsoft não oferece garantia contratual em relação ao software licenciado.
 - Limitação de responsabilidade. Em caso de conduta intencional, negligência grave, reclamações baseadas na Lei de Responsabilidade pelo Produto, bem como, em caso de morte ou lesão corporal, a Microsoft é responsável de acordo com a lei estatutária.

Sujeito à cláusula anterior (ii), a Microsoft somente será responsável por negligência leve se violar tais obrigações contratuais relevantes, cujo cumprimento facilitará a devida execução deste contrato, cuja violação colocaria em risco a finalidade deste contrato e a conformidade com a qual a parte pode confiar constantemente (as chamadas "obrigações cardeais"). Em outros casos de negligência leve, a Microsoft não será responsável.

10 EFEITO JURÍDICO. Este acordo descreve determinados direitos legais. Você pode ter outros direitos sob as leis do seu estado ou país. Este contrato não altera seus direitos sob as leis do seu estado ou país se essas leis não permitirem que isso aconteça. Sem limitação do precedente, para a Austrália, VOCÊ TEM GARANTIAS ESTATUTÁRIAS SOB A LEI DO CONSUMIDOR AUSTRALIANA E NADA NESTES TERMOS PRETENDE AFETAR ESSES DIREITOS

11 ISENÇÃO DE GARANTIA. O SOFTWARE É LICENCIADO NO "ESTADO EM QUE SE ENCONTRA". VOCÊ ARCA COM O RISCO DE USÁ-LO. A MICROSOFT NÃO OFERECE GARANTIAS OU CONDIÇÕES EXPRESSAS. DESDE QUE PERMITIDO PELAS LEIS LOCAIS, A MICROSOFT EXCLUI AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA E NÃO VIOLAÇÃO.

12 LIMITAÇÃO E EXCLUSÃO DE DANOS. VOCÊ PODE RECUPERAR DA MICROSOFT E SEUS FORNECEDORES SOMENTE DANOS DIRETOS ATÉ US\$ 5,00. É VEDADO PLEITEAR INDENIZAÇÃO POR QUAISQUER OUTROS DANOS, INCLUINDO LUCROS CESSANTES, DANOS CONSEQUENTES, ESPECIAIS, INDIRETOS OU INCIDENTAIS.

Essa limitação se aplica a (a) qualquer assunto relacionado ao software, serviços, conteúdo (incluindo código) em sites da Internet de terceiros ou aplicativos de terceiros; e (b) reclamações por violação de contrato, violação de garantia ou condição, responsabilidade estrita, negligência ou outro ato ilícito na medida permitida pela lei aplicável.

Também se aplica mesmo que a Microsoft saiba ou deva saber sobre a possibilidade dos danos. A limitação ou exclusão acima pode não se aplicar a você porque seu país pode não permitir a exclusão ou limitação de danos incidentais, consequentes ou outros.

11) Imagem do Docker baseada em Windows — 3-ondemand-package.cab microsoft-windows-netfx

TERMOS DE LICENÇA ADICIONAIS DO SOFTWARE MICROSOFT

MICROSOFT .NET FRAMEWORK 3.5 SP1 PARA SISTEMA OPERACIONAL MICROSOFT WINDOWS

A Microsoft Corporation (ou uma de suas afiliadas, com base onde você mora) licencia este suplemento para você. Se você estiver licenciado para usar o software do sistema operacional Microsoft Windows, para o qual este suplemento é aplicável, (o "software"), poderá usar este suplemento. Não é possível usá-lo se você não tiver uma licença para o software. Você pode usar uma cópia deste suplemento com cada cópia licenciada válida do software.

Os seguintes termos de licença descrevem termos de uso adicionais para este suplemento. Estes termos e os termos de licença do software se aplicam ao uso do suplemento. Se houver um conflito, esses termos de licença adicionais serão aplicados.

AO USAR ESTE SUPLEMENTO, VOCÊ ACEITA ESTES TERMOS. SE VOCÊ NÃO OS ACEITAR, NÃO USE ESTE SUPLEMENTO.

Se estiver em conformidade com estes termos de licença, você possui os direitos abaixo.

1. SERVIÇOS DE SUPORTE PARA SUPLEMENTO. A Microsoft fornece serviços de suporte para esse software conforme descrito em www.support.microsoft.com/common/international.aspx.
2. TESTE DE COMPARAÇÃO DO MICROSOFT .NET. O software inclui os componentes .NET Framework, Windows Communication Foundation, Windows Presentation Foundation e Windows Workflow Foundation dos sistemas operacionais Windows (componentes .NET). Você pode realizar testes de comparação internos dos Componentes .NET. Você pode divulgar os resultados de qualquer teste de comparação dos Componentes .NET, desde que cumpra as condições estabelecidas em <http://go.microsoft.com/fwlink/?LinkID=66406>.

Não obstante qualquer outro contrato que você possa ter com a Microsoft, se você divulgar tais resultados de testes de comparação, a Microsoft terá o direito de divulgar os resultados dos testes de comparação que realiza de seus produtos que competem com o Componente .NET aplicável, desde que cumpram as mesmas condições estabelecidas em <http://go.microsoft.com/fwlink/?LinkID=66406>.

12) Imagem do Docker baseada em Windows: dotnet-sdk

(disponível em <https://github.com/dotnet/core/blob/main/LICENSE.TXT>)

A licença do MIT (MIT)

Copyright (c) Microsoft Corporation

É concedida permissão, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar com o Software sem restrições, incluindo, sem limitação, os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, além de permitir que as pessoas para as quais o Software é fornecido façam isso, contanto que as seguintes condições sejam atendidas:

O aviso de direitos autorais acima e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, ENTRE OUTRAS, GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM PROPÓSITO ESPECÍFICO E NÃO VIOLAÇÃO. EM NENHUMA CIRCUNSTÂNCIA, OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS DEVEM SER RESPONSABILIZADOS POR QUALQUER RECLAMAÇÃO, DANO OU OUTRA OBRIGAÇÃO, SEJA EM CASO DE AÇÃO CONTRATUAL OU OUTRO ATO

ILÍCITO PROVENIENTE DE OU ASSOCIADO AO SOFTWARE, AO USO OU A OUTROS PROCEDIMENTOS NO SOFTWARE.

Use chaves de CodeBuild condição como variáveis de função de serviço do IAM para controlar o acesso à compilação

Com o ARN de CodeBuild compilação, você pode restringir o acesso aos recursos de compilação usando chaves de contexto para reduzir o acesso aos recursos em sua função de CodeBuild serviço. Pois CodeBuild, as chaves que podem ser usadas para controlar o comportamento de acesso à compilação são `codebuild:buildArn` e `codebuild:projectArn`. Com o ARN do projeto de compilação, você pode verificar se uma chamada para seu recurso veio de um projeto de compilação específico. Para verificar isso, use as chaves de `codebuild:projectArn` condição `codebuild:buildArn` ou em uma política baseada em identidade do IAM.

Para usar as chaves de `codebuild:projectArn` condição `codebuild:buildArn` ou em sua política, inclua-as como uma condição com qualquer um dos operadores de condição do ARN. O valor da chave deve ser uma variável do IAM que é resolvida como um ARN válido. No exemplo de política abaixo, o único acesso permitido será ao projeto de construção com o ARN do projeto para a variável `${codebuild:projectArn}` IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/${codebuild:projectArn}/*"
    }
  ]
}
```

AWS CodeBuild chaves de condição

AWS CodeBuild fornece um conjunto de chaves de condição que você pode usar nas políticas do IAM para aplicar suas políticas organizacionais em CodeBuild recursos como projetos e frotas. As chaves de condição abrangem a maioria dos contextos de solicitação da CodeBuild API, incluindo configurações de rede, configurações de credenciais e restrições computacionais.

Tópicos

- [Aplique as configurações de conectividade VPC em seus projetos e frotas](#)
- [Evite modificações não autorizadas no buildspec do projeto](#)
- [Restrinja os tipos de computação para suas compilações](#)
- [Configurações variáveis do ambiente de controle](#)
- [Use variáveis em nomes de chaves de condição](#)
- [Verifique a existência de atributos nas solicitações de API](#)

Aplique as configurações de conectividade VPC em seus projetos e frotas

Essa política permite que o chamador use os grupos selecionados VPCs, sub-redes e segurança ao criar CodeBuild projetos e frotas. Para obter mais informações sobre chaves de contexto de valores múltiplos, consulte Chaves de contexto de [valor único versus chaves de contexto de valores múltiplos](#).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "codebuild:CreateProject",
      "codebuild:CreateFleet"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "codebuild:vpcConfig.vpcId": [
          "vpc-01234567890abcdef",
          "vpc-abcdef01234567890"
        ],
        "codebuild:vpcConfig.subnets": [
          "subnet-1234abcd",
          "subnet-5678abcd"
        ],
        "codebuild:vpcConfig.securityGroupIds": [
          "sg-12345678abcdefghij",
          "sg-01234567abcdefghij"
        ]
      }
    }
  ]
}
```

```
    }
  }]
}
```

Evite modificações não autorizadas no buildspec do projeto

Essa política não permite que o chamador substitua a especificação de construção no campo. `buildspecOverride`

Note

A chave de `codebuild:source.buildspec` condição suporta somente o operador Null para verificar a existência do campo da API. Ele não avalia o conteúdo do buildspec.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "codebuild:StartBuild",
    "Resource": "*"
  }, {
    "Effect": "Deny",
    "Action": "codebuild:StartBuild",
    "Resource": "*",
    "Condition": {
      "Null": {
        "codebuild:source.buildspec": "false"
      }
    }
  }]
}
```

Restrinja os tipos de computação para suas compilações

Essa política permite criar frotas que podem ser criadas somente com o tipo de [instância de m5.large computação c5.large](#) ou com ele.

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

```

    "Statement": [{
      "Effect": "Allow",
      "Action": "codebuild:CreateFleet",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "codebuild:computeConfiguration.instanceType": ["c5.large", "m5.large"]
        }
      }
    }]
  }
}

```

Configurações variáveis do ambiente de controle

Essa política permite que o chamador substitua a variável de STAGE ambiente para ser ouBETA. GAMMA Também nega explicitamente a substituição de ser PRODUCTION e nega STAGE a substituição da variável de ambiente. MY_APP_VERSION Para chaves de contexto de vários valores, consulte Chaves de contexto [de valor único versus chaves de contexto de valores múltiplos](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "codebuild:environment.environmentVariables/STAGE.value": [
            "BETA",
            "GAMMA"
          ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:StartBuild"
      ],
      "Resource": "*",
    }
  ]
}

```

```

        "Condition": {
            "StringEquals": {
                "codebuild:environment.environmentVariables/STAGE.value":
"PRODUCTION"
            },
            "ForAnyValue:StringEquals": {
                "codebuild:environment.environmentVariables.name": [
                    "MY_APP_VERSION"
                ]
            }
        }
    }
}

```

Use variáveis em nomes de chaves de condição

Você pode usar variáveis em nomes de chaves de condição, como `secondarySources/${sourceIdentifier}.location` e `secondaryArtifacts/${artifactIdentifier}.location`, onde você pode especificar sua [fonte](#) secundária ou identificador de [artefato](#) secundário na política do IAM. A política abaixo permite que o chamador crie um projeto com um local de origem específico para a fonte `mySecondSource` secundária.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "codebuild:CreateProject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codebuild:secondarySources/mySecondSource.location": "my-source-
location"
                }
            }
        }
    ]
}

```


Verifique a existência de atributos nas solicitações de API

CodeBuild suporta chaves de condição para verificar a existência de alguns campos na solicitação da API. A política impõe o requisito de VPC ao criar ou atualizar projetos.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "codebuild:CreateProject",
      "codebuild:UpdateProject"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "codebuild:vpcConfig": "false"
      }
    }
  }]
}
```

Exemplos de código para CodeBuild usar AWS SDKs

Os exemplos de código a seguir mostram como usar CodeBuild com um kit AWS de desenvolvimento de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- [Exemplos básicos de CodeBuild uso AWS SDKs](#)
 - [Ações para CodeBuild usar AWS SDKs](#)
 - [Use CreateProject com um AWS SDK ou CLI](#)
 - [Use ListBuilds com um AWS SDK ou CLI](#)
 - [Use ListProjects com um AWS SDK ou CLI](#)
 - [Use StartBuild com um AWS SDK ou CLI](#)

Exemplos básicos de CodeBuild uso AWS SDKs

Os exemplos de código a seguir mostram como usar o básico do AWS CodeBuild with AWS SDKs.

Exemplos

- [Ações para CodeBuild usar AWS SDKs](#)
 - [Use CreateProject com um AWS SDK ou CLI](#)
 - [Use ListBuilds com um AWS SDK ou CLI](#)
 - [Use ListProjects com um AWS SDK ou CLI](#)
 - [Use StartBuild com um AWS SDK ou CLI](#)

Ações para CodeBuild usar AWS SDKs

Os exemplos de código a seguir demonstram como realizar CodeBuild ações individuais com AWS SDKs. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a [Referência de APIs do AWS CodeBuild](#).

Exemplos

- [Use CreateProject com um AWS SDK ou CLI](#)
- [Use ListBuilds com um AWS SDK ou CLI](#)
- [Use ListProjects com um AWS SDK ou CLI](#)
- [Use StartBuild com um AWS SDK ou CLI](#)

Use **CreateProject** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o CreateProject.

CLI

AWS CLI

Exemplo 1: Para criar um projeto de AWS CodeBuild construção

O create-project exemplo a seguir cria um projeto de CodeBuild compilação usando arquivos de origem de um bucket do S3.

```
aws codebuild create-project \
  --name "my-demo-project" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
input-bucket/my-source.zip\"}" \
  --artifacts "{\"type\": \"S3\", \"location\": \"codebuild-us-  
west-2-123456789012-output-bucket\"}" \
  --environment "{\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/  
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role"
```

Saída:

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "name": "my-cli-demo-project",
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
    "lastModified": 1556839783.274,
    "badge": {
      "badgeEnabled": false
    },
    "queuedTimeoutInMinutes": 480,
    "environment": {
      "image": "aws/codebuild/standard:1.0",
      "computeType": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "imagePullCredentialsType": "CODEBUILD",
      "privilegedMode": false,
      "environmentVariables": []
    },
    "artifacts": {
      "location": "codebuild-us-west-2-123456789012-output-bucket",
      "name": "my-cli-demo-project",
      "namespaceType": "NONE",
      "type": "S3",
      "packaging": "NONE",
      "encryptionDisabled": false
    },
    "source": {
      "type": "S3",
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip",
      "insecureSsl": false
    },
    "timeoutInMinutes": 60,
    "cache": {
      "type": "NO_CACHE"
    },
    "created": 1556839783.274
  }
}
```

```
}
```

Exemplo 2: Para criar um projeto de AWS CodeBuild construção usando um arquivo de entrada JSON para os parâmetros

O `create-project` exemplo a seguir cria um projeto de CodeBuild construção passando todos os parâmetros necessários em um arquivo de entrada JSON. Crie o modelo do arquivo de entrada executando o comando somente com o `--generate-cli-skeleton` `parameter`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

O arquivo de entrada JSON `create-project.json` contém o seguinte conteúdo:

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:1.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "serviceIAMRole"
}
```

Saída:

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
```

```
        "location": "codebuild-region-ID-account-ID-output-bucket",
        "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
        "computeType": "BUILD_GENERAL1_SMALL",
        "image": "aws/codebuild/standard:1.0",
        "type": "LINUX_CONTAINER",
        "environmentVariables": []
    },
    "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
    },
    "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
    }
}
```

Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#) no Guia do AWS CodeBuild usuário.

- Para obter detalhes da API, consulte [CreateProject](#) em Referência de AWS CLI Comandos.

JavaScript

SDK para JavaScript (v3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie um projeto.

```
import {
    ArtifactsType,
```

```
CodeBuildClient,
ComputeType,
CreateProjectCommand,
EnvironmentType,
SourceType,
} from "@aws-sdk/client-codebuild";

// Create the AWS CodeBuild project.
export const createProject = async (
  projectName = "MyCodeBuilder",
  roleArn = "arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin",
  buildOutputBucket = "xxxx",
  githubUrl = "https://...",
) => {
  const codeBuildClient = new CodeBuildClient({});

  const response = await codeBuildClient.send(
    new CreateProjectCommand({
      artifacts: {
        // The destination of the build artifacts.
        type: ArtifactsType.S3,
        location: buildOutputBucket,
      },
      // Information about the build environment. The combination of
      "computeType" and "type" determines the
      // requirements for the environment such as CPU, memory, and disk space.
      environment: {
        // Build environment compute types.
        // https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-compute-types.html
        computeType: ComputeType.BUILD_GENERAL1_SMALL,
        // Docker image identifier.
        // See https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-available.html
        image: "aws/codebuild/standard:7.0",
        // Build environment type.
        type: EnvironmentType.LINUX_CONTAINER,
      },
      name: projectName,
      // A role ARN with permission to create a CodeBuild project, write to the
      artifact location, and write CloudWatch logs.
      serviceRole: roleArn,
      source: {
        // The type of repository that contains the source code to be built.
```

```

        type: SourceType.GITHUB,
        // The location of the repository that contains the source code to be
built.
        location: githubUrl,
    },
    )),
);
console.log(response);
// {
//   '$metadata': {
//     httpStatusCode: 200,
//     requestId: 'b428b244-777b-49a6-a48d-5dffedced8e7',
//     extendedRequestId: undefined,
//     cfId: undefined,
//     attempts: 1,
//     totalRetryDelay: 0
//   },
//   project: {
//     arn: 'arn:aws:codebuild:us-east-1:xxxxxxxxxxxx:project/MyCodeBuilder',
//     artifacts: {
//       encryptionDisabled: false,
//       location: 'xxxxxx-xxxxxx-xxxxxx',
//       name: 'MyCodeBuilder',
//       namespaceType: 'NONE',
//       packaging: 'NONE',
//       type: 'S3'
//     },
//     badge: { badgeEnabled: false },
//     cache: { type: 'NO_CACHE' },
//     created: 2023-08-18T14:46:48.979Z,
//     encryptionKey: 'arn:aws:kms:us-east-1:xxxxxxxxxxxx:alias/aws/s3',
//     environment: {
//       computeType: 'BUILD_GENERAL1_SMALL',
//       environmentVariables: [],
//       image: 'aws/codebuild/standard:7.0',
//       imagePullCredentialsType: 'CODEBUILD',
//       privilegedMode: false,
//       type: 'LINUX_CONTAINER'
//     },
//     lastModified: 2023-08-18T14:46:48.979Z,
//     name: 'MyCodeBuilder',
//     projectVisibility: 'PRIVATE',
//     queuedTimeoutInMinutes: 480,
//     serviceRole: 'arn:aws:iam:xxxxxxxxxxxx:role/CodeBuildAdmin',

```



```
//      source: {  
//          insecureSsl: false,  
//          location: 'https://...',  
//          reportBuildStatus: false,  
//          type: 'GITHUB'  
//      },  
//      timeoutInMinutes: 60  
//  }  
// }  
return response;  
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK para JavaScript](#).
- Para obter detalhes da API, consulte [CreateProject](#) a Referência AWS SDK para JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListBuilds** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o ListBuilds.

C++

SDK para C++

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
//! List the CodeBuild builds.  
/*!  
    \param sortType: 'SortOrderType' type.  
    \param clientConfiguration: AWS client configuration.
```

```

    \return bool: Function succeeded.
    */
bool AwsDoc::CodeBuild::listBuilds(Aws::CodeBuild::Model::SortOrderType sortType,
                                   const Aws::Client::ClientConfiguration
                                   &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListBuildsRequest listBuildsRequest;
    listBuildsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Used for pagination.

    do {
        if (!nextToken.empty()) {
            listBuildsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListBuildsOutcome listBuildsOutcome =
codeBuildClient.ListBuilds(
    listBuildsRequest);

        if (listBuildsOutcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &ids =
listBuildsOutcome.GetResult().GetIds();
            if (!ids.empty()) {

                std::cout << "Information about each build:" << std::endl;
                Aws::CodeBuild::Model::BatchGetBuildsRequest getBuildsRequest;
                getBuildsRequest.SetIds(listBuildsOutcome.GetResult().GetIds());
                Aws::CodeBuild::Model::BatchGetBuildsOutcome getBuildsOutcome =
codeBuildClient.BatchGetBuilds(
                    getBuildsRequest);

                if (getBuildsOutcome.IsSuccess()) {
                    const Aws::Vector<Aws::CodeBuild::Model::Build> &builds =
getBuildsOutcome.GetResult().GetBuilds();
                    std::cout << builds.size() << " build(s) found." <<
std::endl;

                    for (auto val: builds) {
                        std::cout << val.GetId() << std::endl;
                    }
                } else {
                    std::cerr << "Error getting builds"

```

```
        << getBuildsOutcome.GetError().GetMessage() <<
std::endl;
        return false;
    }
} else {
    std::cout << "No builds found." << std::endl;
}

// Get the next token for pagination.

nextToken = listBuildsOutcome.GetResult().GetNextToken();
} else {
    std::cerr << "Error listing builds"
        << listBuildsOutcome.GetError().GetMessage()
        << std::endl;
    return false;
}

} while (!nextToken.

    empty()

);

return true;
}
```

- Para obter detalhes da API, consulte [ListBuilds](#) na Referência AWS SDK para C++ da API.

CLI

AWS CLI

Para obter uma lista de AWS CodeBuild construções. IDs

O `list-builds` exemplo a seguir obtém uma lista de CodeBuild IDs classificados em ordem crescente.

```
aws codebuild list-builds --sort-order ASCENDING
```

A saída inclui um valor `nextToken` que indica que há mais saída disponível.

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for
  brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Execute esse comando novamente e forneça o valor `nextToken` na resposta anterior como parâmetro para obter a próxima parte da saída. Repita até que você não receba um valor `nextToken` na resposta.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Próxima parte da saída:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Para obter mais informações, consulte [Exibir uma lista de IDs compilações \(AWS CLI\) no Guia do usuário AWS CodeBuild](#)

- Para obter detalhes da API, consulte [ListBuilds](#) em Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListProjects** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListProjects`.

C++

SDK para C++

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
//! List the CodeBuild projects.
/*!
    \param sortType: 'SortOrderType' type.
    \param clientConfiguration: AWS client configuration.
    \return bool: Function succeeded.
*/
bool AwsDoc::CodeBuild::listProjects(Aws::CodeBuild::Model::SortOrderType
    sortType,
                                     const Aws::Client::ClientConfiguration
    &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListProjectsRequest listProjectsRequest;
    listProjectsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Next token for pagination.
    Aws::Vector<Aws::String> allProjects;

    do {
        if (!nextToken.empty()) {
            listProjectsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListProjectsOutcome outcome =
        codeBuildClient.ListProjects(
            listProjectsRequest);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &projects =
            outcome.GetResult().GetProjects();
            allProjects.insert(allProjects.end(), projects.begin(),
            projects.end());
        }
    } while (nextToken != "");
}
```

```
        nextToken = outcome.GetResult().GetNextToken();
    }

    else {
        std::cerr << "Error listing projects" <<
outcome.GetError().GetMessage()
        << std::endl;
    }

    } while (!nextToken.empty());

    std::cout << allProjects.size() << " project(s) found." << std::endl;
    for (auto project: allProjects) {
        std::cout << project << std::endl;
    }

    return true;
}
```

- Para obter detalhes da API, consulte [ListProjects](#) a Referência AWS SDK para C++ da API.

CLI

AWS CLI

Para obter uma lista dos nomes dos projetos de AWS CodeBuild construção.

O list-projects exemplo a seguir obtém uma lista de projetos de CodeBuild construção classificados por nome em ordem crescente.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

A saída inclui um valor nextToken que indica que há mais saída disponível.

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
```

```
        ... The full list of build project names has been omitted for
    brevity ...
        "codebuild-demo-project99"
    ]
}
```

Execute esse comando novamente e forneça o valor `nextToken` da resposta anterior como parâmetro para obter a próxima parte da saída. Repita até que você não receba um valor `nextToken` na resposta.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",

    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"

  ]
}
```

Para obter mais informações, consulte [Exibir uma lista de nomes de projetos de construção \(AWS CLI\)](#) no Guia do AWS CodeBuild usuário.

- Para obter detalhes da API, consulte [ListProjects](#) em Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **StartBuild** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `StartBuild`.

C++

SDK para C++

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#!/ Start an AWS CodeBuild project build.
/*!
  \param projectName: A CodeBuild project name.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::startBuild(const Aws::String &projectName,
                                   const Aws::Client::ClientConfiguration
&clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::StartBuildRequest startBuildRequest;
    startBuildRequest.SetProjectName(projectName);

    Aws::CodeBuild::Model::StartBuildOutcome outcome =
codeBuildClient.StartBuild(
    startBuildRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully started build" << std::endl;
        std::cout << "Build ID: " << outcome.GetResult().GetBuild().GetId()
            << std::endl;
    }

    else {
        std::cerr << "Error starting build" << outcome.GetError().GetMessage()
            << std::endl;
    }

    return outcome.IsSuccess();
}
```


- Para obter detalhes da API, consulte [StartBuild](#) da Referência AWS SDK para C++ da API.

CLI

AWS CLI

Para começar a executar uma compilação de um projeto de AWS CodeBuild compilação.

O `start-build` exemplo a seguir inicia uma compilação para o CodeBuild projeto especificado. A compilação substitui a configuração do projeto pelo número de minutos que a compilação pode ficar na fila antes de atingir o tempo limite e as configurações de artefato do projeto.

```
aws codebuild start-build \  
  --project-name "my-demo-project" \  
  --queued-timeout-in-minutes-override 5 \  
  --artifacts-override {"\"type\": \"S3\", \"location\": \  
  \"arn:aws:s3:::artifacts-override\", \"overrideArtifactName\": true"}
```

Saída:

```
{  
  "build": {  
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",  
    "buildStatus": "IN_PROGRESS",  
    "buildComplete": false,  
    "projectName": "my-demo-project",  
    "timeoutInMinutes": 60,  
    "source": {  
      "insecureSsl": false,  
      "type": "S3",  
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip"  
    },  
    "queuedTimeoutInMinutes": 5,  
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",  
    "currentPhase": "QUEUED",  
    "startTime": 1556905683.568,  
    "environment": {  
      "computeType": "BUILD_GENERAL1_MEDIUM",  
      "environmentVariables": [],  

```

```

        "type": "LINUX_CONTAINER",
        "privilegedMode": false,
        "image": "aws/codebuild/standard:1.0",
        "imagePullCredentialsType": "CODEBUILD"
    },
    "phases": [
        {
            "phaseStatus": "SUCCEEDED",
            "startTime": 1556905683.568,
            "phaseType": "SUBMITTED",
            "durationInSeconds": 0,
            "endTime": 1556905684.524
        },
        {
            "startTime": 1556905684.524,
            "phaseType": "QUEUED"
        }
    ],
    "logs": {
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=null;stream=null"
    },
    "artifacts": {
        "encryptionDisabled": false,
        "location": "arn:aws:s3:::artifacts-override/my-demo-project",
        "overrideArtifactName": true
    },
    "cache": {
        "type": "NO_CACHE"
    },
    "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
    "initiator": "my-aws-account-name",
    "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
    }
}

```

Para obter mais informações, consulte [Executar uma compilação \(AWS CLI\)](#) no Guia do AWS CodeBuild usuário.

- Para obter detalhes da API, consulte [StartBuild](#) em Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Solução de problemas AWS CodeBuild

Use as informações neste tópico para ajudá-lo a identificar, diagnosticar e resolver problemas. Para saber como registrar e monitorar CodeBuild compilações para solucionar problemas, consulte.

[Registro em log e monitoramento](#)

Tópicos

- [Artefatos de referência de compilações Apache Maven do repositório errado](#)
- [Comandos de compilação executados como raiz por padrão](#)
- [As compilações podem falhar quando nomes de arquivos têm caracteres que não sejam do inglês.](#)
- [As compilações podem falhar ao obter parâmetros do Amazon EC2 Parameter Store](#)
- [Não é possível acessar o filtro da ramificação de acesso no console do CodeBuild](#)
- [Não é possível visualizar o êxito ou a falha de compilação](#)
- [Status de compilação não comunicado ao provedor de origem](#)
- [Não é possível localizar e selecionar a imagem de base da plataforma Windows Server Core 2019.](#)
- [Comandos anteriores em arquivos buildspec não são reconhecidos por comandos posteriores](#)
- [Erro: "acesso negado" ao tentar fazer download do cache](#)
- [Erro: "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" ao usar uma imagem de compilação personalizada](#)
- [Erro: "O contêiner de compilação foi encontrado inativo antes de concluir a compilação. O contêiner de compilação morreu porque estava sem memória ou a imagem do Docker não é suportada. ErrorCode: 500"](#)
- [Erro: "Cannot connect to the Docker daemon \(Não é possível conectar-se ao daemon do Docker\)" ao executar uma compilação](#)
- [Erro: "não CodeBuild está autorizado a executar: sts:AssumeRole" ao criar ou atualizar um projeto de compilação](#)
- [Erro: "Erro ao chamar GetBucketAcl: ou o proprietário do bucket mudou ou a função de serviço não tem mais permissão para chamar s3:GetBucketAcl"](#)
- [Erro: "Failed to upload artifacts: Invalid arn \(Falha ao fazer upload de artefatos: arn inválido\)" ao executar uma compilação](#)
- [Erro: "falha do clone do Git: não é possível acessar 'your-repository-URL': problema de certificado SSL: certificado autoassinado"](#)

- [Erro: "The bucket you are attempting to access must be addressed using the specified endpoint \(O bucket que você está tentando acessar deve ser endereçado usando o endpoint especificado\)" ao executar uma compilação](#)
- [Erro: "This build image requires selecting at least one runtime version" \(Esta imagem de compilação requer a seleção de pelo menos um tempo de execução\)](#)
- [Erro: "QUEUED: INSUFFICIENT_SUBNET" quando ocorre uma falha em uma compilação em uma fila de compilação](#)
- [Erro: "Não foi possível baixar o cache: RequestError: Falha na solicitação de envio causada por: x509: Falha ao carregar raízes do sistema e nenhuma raiz fornecida"](#)
- [Erro: "Não foi possível baixar o certificado do S3. AccessDenied"](#)
- [Erro: "não foi possível localizar as credenciais"](#)
- [RequestError erro de tempo limite ao executar CodeBuild em um servidor proxy](#)
- [O bourne shell \(sh\) deve existir em imagens de compilação](#)
- [Aviso: "ignorando a instalação de tempos de execução. A seleção de versão de tempo de execução não é compatível com esta imagem de compilação" ao executar uma compilação](#)
- [Erro: "Não foi possível verificar a JobWorker identidade" ao abrir o CodeBuild console](#)
- [Falha ao iniciar a compilação](#)
- [Acessando GitHub metadados em compilações armazenadas em cache localmente](#)
- [AccessDenied: O proprietário do bucket do grupo de relatórios não corresponde ao proprietário do bucket do S3...](#)
- [Erro: "Suas credenciais não têm um ou mais escopos de privilégios necessários" ao criar um projeto com CodeBuild CodeConnections](#)
- [Erro: "Desculpe, nenhum terminal foi solicitado - não foi possível obter a entrada" ao compilar com o comando de instalação do Ubuntu](#)

Artefatos de referência de compilações Apache Maven do repositório errado

Problema: [quando você usa o Maven com um ambiente AWS CodeBuild de compilação Java fornecido, o Maven extrai dependências de compilação e plug-in do repositório central seguro do Maven em <https://repo1.maven.org/maven2>](#). Isso ocorre mesmo quando o arquivo `pom.xml` do seu projeto de build declara explicitamente outros locais de uso.

Possível causa: os ambientes CodeBuild de compilação Java fornecidos incluem um arquivo chamado `settings.xml` que está pré-instalado no diretório do `/root/.m2` ambiente de compilação. Esse arquivo `settings.xml` contém as seguintes declarações, que instruem o Maven para sempre chamar as dependências de plugin e build do repositório central seguro Maven em <https://repo1.maven.org/maven2>.

```
<settings>
  <activeProfiles>
    <activeProfile>securecentral</activeProfile>
  </activeProfiles>
  <profiles>
    <profile>
      <id>securecentral</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

Solução recomendada: Faça o seguinte:

1. Adicione um arquivo `settings.xml` ao seu código-fonte.
2. Nesse arquivo `settings.xml`, use o formato `settings.xml` precedente como guia para declarar os repositórios de onde deseja que o Maven chame as dependências de plugin e build.

3. Na `install` fase do seu projeto de compilação, CodeBuild instrua a copiar seu `settings.xml` arquivo para o `/root/.m2` diretório do ambiente de compilação. Por exemplo, considere o seguinte trecho de um arquivo `buildspec.yml` que demonstre esse comportamento.

```
version 0.2

phases:
  install:
    commands:
      - cp ./settings.xml /root/.m2/settings.xml
```

Comandos de compilação executados como raiz por padrão

Problema: AWS CodeBuild executa seus comandos de compilação como usuário `root`. Isso acontecerá mesmo se o `Dockerfile` da imagem de compilação relacionado definir a instrução `USER` como um usuário diferente.

Causa: Por padrão, CodeBuild executa todos os comandos de compilação como usuário `root`.

Solução recomendada: nenhuma.

As compilações podem falhar quando nomes de arquivos têm caracteres que não sejam do inglês.

Problema: ao executar uma compilação que use arquivos com nomes que contenham caracteres que não sejam do inglês (por exemplo, caracteres chineses), a compilação falha.

Possível causa: ambientes de compilação fornecidos por AWS CodeBuild têm sua localidade padrão definida como `POSIX`. As configurações de localização `POSIX` são menos compatíveis com nomes CodeBuild de arquivos que não sejam dos EUA. Caracteres em inglês e podem fazer com que compilações relacionadas falhem.

Solução recomendada: adicione os comandos a seguir à seção `pre_build` do arquivo `buildspec`. Esses comandos fazem com que o ambiente de compilação use UTF-8 em inglês dos EUA para suas configurações de localização, que é mais compatível com CodeBuild nomes de arquivos que não sejam dos EUA. Caracteres em inglês.

Para ambientes de compilação com base no Ubuntu:

```
pre_build:
  commands:
    - export LC_ALL="en_US.UTF-8"
    - locale-gen en_US en_US.UTF-8
    - dpkg-reconfigure -f noninteractive locales
```

Para ambientes de compilação com base no Amazon Linux:

```
pre_build:
  commands:
    - export LC_ALL="en_US.utf8"
```

As compilações podem falhar ao obter parâmetros do Amazon EC2 Parameter Store

Problema: quando uma compilação tenta obter o valor de um ou mais parâmetros armazenados no Amazon EC2 Parameter Store, a compilação falha na `DOWNLOAD_SOURCE` fase com o erro `Parameter does not exist`.

Causa possível: a função de serviço da qual o projeto de construção depende não tem permissão para chamar a `ssm:GetParameters` ação ou o projeto de construção usa uma função de serviço gerada AWS CodeBuild e que permite chamar a `ssm:GetParameters` ação, mas os parâmetros têm nomes que não começam com `/CodeBuild/`.

Soluções recomendadas:

- Se a função de serviço não foi gerada por CodeBuild, atualize sua definição CodeBuild para permitir chamar a `ssm:GetParameters` ação. Por exemplo, a seguinte declaração de política permite chamar a ação `ssm:GetParameters` para obter parâmetros com nomes que comecem com `/CodeBuild/`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
      "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/CodeBuild/*"
    }
  ]
}
```



```
]
}
```

- Se a função de serviço foi gerada por CodeBuild, atualize sua definição para permitir CodeBuild o acesso aos parâmetros no Amazon EC2 Parameter Store com nomes diferentes daqueles que começam com `/CodeBuild/`. Por exemplo, a seguinte declaração de política permite chamar a ação `ssm:GetParameters` para obter parâmetros com o nome especificado:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
      "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/PARAMETER_NAME"
    }
  ]
}
```

Não é possível acessar o filtro da ramificação de acesso no console do CodeBuild

Problema: a opção de filtro de ramificação não está disponível no console quando você cria ou atualiza um AWS CodeBuild projeto.

Possível causa: a opção de filtro da ramificação está suspensa. Ela foi substituída por grupos de filtros de webhook, que fornecem mais controle sobre os eventos de webhook que acionam uma nova compilação no CodeBuild.

Solução recomendada: para migrar um filtro de ramificação criado antes da introdução dos filtros de webhook, crie um grupo de filtro de webhook com um filtro HEAD_REF com a expressão regular `^refs/heads/branchName$`. Por exemplo, se sua expressão regular do filtro de ramificação era `^branchName$`, então a expressão regular que você colocar no filtro HEAD_REF será `^refs/heads/branchName$`. Para obter mais informações, consulte [Filtrar eventos de webhook do Bitbucket](#) e [Filtrar eventos de GitHub webhook \(console\)](#).

Não é possível visualizar o êxito ou a falha de compilação

Problema: você não consegue ver o êxito ou a falha de uma nova tentativa de compilação.

Possível causa: a opção para informar o status da compilação não está habilitada.

Soluções recomendadas: ative o status de criação do relatório ao criar ou atualizar um CodeBuild projeto. Essa opção instrui o CodeBuild a informar o status quando você acionar uma compilação. Para obter mais informações, consulte [reportBuildStatus](#) na Referência de APIs do AWS CodeBuild .

Status de compilação não comunicado ao provedor de origem

Problema: depois de permitir o relatório do status da compilação para um provedor de origem, como GitHub o Bitbucket, o status da compilação não é atualizado.

Possível causa: o usuário associado ao provedor de origem não tem acesso de gravação ao repositório.

Soluções recomendadas: para poder comunicar o status da compilação ao provedor de origem, o usuário associado ao provedor de origem deve ter acesso de gravação ao repositório. Se o usuário não tiver acesso de gravação, o status de compilação não poderá ser atualizado. Para obter mais informações, consulte [Acesso do provedor de origem](#).

Não é possível localizar e selecionar a imagem de base da plataforma Windows Server Core 2019.

Problema: não é possível localizar nem selecionar a imagem de base da plataforma Windows Server Core 2019.

Possível causa: você está usando uma AWS região que não suporta essa imagem.

Soluções recomendadas: use uma das seguintes regiões da AWS com a qual a imagem base da plataforma Windows Server Core 2019 é compatível:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (Oregon)
- Europa (Irlanda)

Comandos anteriores em arquivos buildspec não são reconhecidos por comandos posteriores

Problema: Os resultados de um ou mais comandos em seu arquivo buildspec não são reconhecidos por comandos posteriores no mesmo arquivo buildspec. Por exemplo, um comando pode definir uma variável de ambiente local, mas um comando executado posteriormente falha em obter o valor da variável de ambiente local.

Possível causa: no arquivo buildspec versão 0.1, o AWS CodeBuild executa cada comando em uma instância à parte do shell padrão no ambiente de compilação. Isso significa que cada comando é executado isoladamente em relação aos outros comandos. Por padrão, então, você não pode executar um comando que dependa do estado de quaisquer comandos anteriores.

Soluções recomendadas: recomendamos usar a especificação de compilação versão 0.2, que resolve o problema. Se você precisa usar o buildspec versão 0.1, recomendamos que use o operador de encadeamento de comandos do shell (por exemplo, `&&` no Linux) para combinar vários comandos em um só. Ou inclua no código-fonte um script de shell que contenha vários comandos e, em seguida, chame o script de shell de um único comando no arquivo buildspec. Para obter mais informações, consulte [Shells e comandos em ambientes de compilação](#) e [Variáveis de ambiente em ambientes de compilação](#).

Erro: "acesso negado" ao tentar fazer download do cache

Problema: ao tentar fazer download do cache em um projeto de compilação que tenha cache habilitado, você recebe um erro `Access denied`.

Causas possíveis:

- Você acabou de configurar o armazenamento em cache como parte do projeto de compilação.
- O cache foi invalidado recentemente por meio da API `InvalidateProjectCache`.
- A função de serviço que está sendo usada por CodeBuild não tem `s3:GetObject` e `s3:PutObject` permissões para o bucket do S3 que contém o cache.

Solução recomendada: para uso pela primeira vez, é normal ver isso logo depois da atualização da configuração do cache. Caso esse erro persista, é necessário verificar se a função de serviço tem permissões `s3:GetObject` e `s3:PutObject` para o bucket do S3 que está mantendo o cache.

Para obter mais informações, consulte [Specifying S3 permissions](#) no Guia do desenvolvedor do Amazon S3.

Erro: "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" ao usar uma imagem de compilação personalizada

Problema: durante a tentativa de executar uma compilação que use uma imagem de compilação personalizada, a compilação falha com o erro BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE.

Possível causa: o tamanho descompactado geral da imagem de compilação é maior que o espaço em disco disponível do tipo de computação do ambiente de compilação. Para verificar o tamanho da imagem da compilação, use o Docker para executar o comando `docker images REPOSITORY:TAG`. Para obter uma lista de espaço em disco disponível por tipo de computação, consulte [Modos e tipos de computação do ambiente de compilação](#).

Solução recomendada: use um tipo de computação maior com mais espaço em disco disponível ou reduza o tamanho da imagem de compilação personalizada.

Possível causa: AWS CodeBuild não tem permissão para extrair a imagem de compilação do seu Amazon Elastic Container Registry (Amazon ECR).

Solução recomendada: atualize as permissões em seu repositório no Amazon ECR para que CodeBuild você possa inserir sua imagem de compilação personalizada no ambiente de compilação. Para obter mais informações, consulte o [Exemplo do Amazon ECR](#).

Possível causa: a imagem do Amazon ECR que você solicitou não está disponível na AWS região que sua AWS conta está usando.

Solução recomendada: use uma imagem do Amazon ECR que esteja na mesma AWS região que a que sua AWS conta está usando.

Possível causa: você está usando um registro privado em uma VPC que não tem acesso público à Internet. CodeBuild não é possível extrair uma imagem de um endereço IP privado em uma VPC. Para obter mais informações, consulte [Registro privado com AWS Secrets Manager amostra para CodeBuild](#).

Solução recomendada: se você usar um registro privado em uma VPC, verifique se a VPC tem acesso público à internet.

Possível causa: se a mensagem de erro contiver “toomanyrequests” e a imagem for obtida do Docker Hub, esse erro significa que o limite de extração do Docker Hub foi atingido.

Solução recomendada: use um registro privado do Docker Hub ou obtenha a imagem do Amazon ECR. Para obter mais informações sobre como usar um registro privado, consulte [Registro privado com AWS Secrets Manager amostra para CodeBuild](#). Para obter mais informações sobre como usar o Amazon ECR, consulte [Exemplo de Amazon ECR para CodeBuild](#).

Erro: “O contêiner de compilação foi encontrado inativo antes de concluir a compilação. O contêiner de compilação morreu porque estava sem memória ou a imagem do Docker não é suportada. ErrorCode: 500”

Problema: quando você tenta usar um contêiner Microsoft Windows ou Linux no AWS CodeBuild, esse erro ocorre durante a fase de PROVISIONAMENTO.

Causas possíveis:

- A versão do sistema operacional do contêiner não é suportada pelo CodeBuild.
- HTTP_PROXY, HTTPS_PROXY ou ambos são especificados no contêiner.

Soluções recomendadas:

- Para o Microsoft Windows, use um contêiner do Windows com um sistema operacional de contêiner (versão: microsoft/windowsservercore:10.0.x (for example, microsoft/windowsservercore 10.0.14393.2125)).
- Para o Linux, desmarque as configurações HTTPS_PROXY e HTTP_PROXY em sua imagem do Docker ou especifique a configuração da VPC no projeto de compilação.

Erro: “Cannot connect to the Docker daemon (Não é possível conectar-se ao daemon do Docker)” ao executar uma compilação

Problema: sua compilação falha e você recebe um erro semelhante a Cannot connect to the Docker daemon at unix:/var/run/docker.sock. Is the docker daemon running? no log da compilação.

Causa possível: sua compilação não está sendo executada no modo privilegiado.

Solução recomendada: para corrigir esse erro, você deve ativar o modo privilegiado e atualizar o buildspec usando as instruções a seguir.

Para executar a compilação no modo privilegiado, siga estas etapas:

1. Abra o CodeBuild console em <https://console.aws.amazon.com/codebuild/>.
2. No painel de navegação, escolha Projetos de compilação e selecione o projeto de compilação.
3. Em Edit (Editar), escolha Environment (Ambiente).
4. Escolha Additional configuration (Configuração adicional).
5. Em Privilegiado, selecione Ativar este sinalizador se quiser criar imagens do Docker ou desejar que suas compilações obtenham privilégios elevados.
6. Selecione Update environment (Atualizar ambiente).
7. Selecione Start build (Iniciar compilação) para tentar novamente.

Você também precisará iniciar o daemon do Docker dentro do contêiner. A fase `install` do buildspec pode ser semelhante a esta.

```
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Para obter mais informações sobre o driver de armazenamento OverlayFS mencionado no arquivo buildspec, consulte [Usar o driver de armazenamento OverlayFS](#) no site do Docker.

Note

Caso o sistema operacional base seja o Alpine Linux, no arquivo `buildspec.yml` adicione o argumento `-t` em `timeout`:

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Para saber mais sobre como criar e executar uma imagem do Docker usando AWS CodeBuild, consulte [Docker em amostra de imagem personalizada para CodeBuild](#).

Erro: "não CodeBuild está autorizado a executar: sts:AssumeRole" ao criar ou atualizar um projeto de compilação

Problema: ao tentar criar ou atualizar um projeto de compilação, você recebe o erro `Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name`.

Causas possíveis:

- O AWS Security Token Service (AWS STS) foi desativado para a AWS região em que você está tentando criar ou atualizar o projeto de compilação.
- A função AWS CodeBuild de serviço associada ao projeto de compilação não existe ou não tem permissões suficientes para confiar CodeBuild.
- O caso da função de AWS CodeBuild serviço associado ao projeto de compilação não corresponde à função real do IAM.

Soluções recomendadas:

- Certifique-se de que AWS STS está ativado para a AWS região em que você está tentando criar ou atualizar o projeto de compilação. Para obter mais informações, consulte [Ativação e desativação AWS STS em uma AWS região no Guia](#) do usuário do IAM.
- Verifique se a função CodeBuild de serviço de destino existe em sua AWS conta. Se você não estiver usando o console, certifique-se de que não soletrou erradamente o Amazon Resource Name (ARN) da função de serviço quando criou ou atualizou o projeto de build. Observe que as funções do IAM diferenciam maiúsculas de minúsculas, portanto, verifique se as maiúsculas e minúsculas da função do IAM estão corretas.
- Certifique-se de que a função CodeBuild de serviço de destino tenha permissões suficientes para confiar CodeBuild. Para obter mais informações, consulte a declaração de políticas de relacionamento de confiança em [CodeBuild Permitir interagir com outros AWS serviços](#).

Erro: “Erro ao chamar GetBucketAcl: ou o proprietário do bucket mudou ou a função de serviço não tem mais permissão para chamar s3:GetBucketAcl”

Problema: ao executar uma compilação, você recebe um erro sobre a mudança de propriedade de um bucket do S3 e as permissões GetBucketAcl.

Possível causa: você adicionou as permissões s3:GetBucketAcl e s3:GetBucketLocation ao perfil do IAM. Essas permissões protegem o bucket do S3 de seu projeto e garantem que só você pode acessá-lo. Depois de adicionar essas permissões, o proprietário do bucket do S3 foi alterado.

Solução recomendada: verifique se você é proprietário do bucket do S3 e adicione permissões ao perfil do IAM novamente. Para obter mais informações, consulte [Acesso seguro aos buckets do S3](#).

Erro: "Failed to upload artifacts: Invalid arn (Falha ao fazer upload de artefatos: arn inválido)" ao executar uma compilação

Problema: ao executar uma compilação, a fase UPLOAD_ARTIFACTS de compilação falha com o erro Failed to upload artifacts: Invalid arn.

Possível causa: seu bucket de saída do S3 (o bucket onde AWS CodeBuild armazena a saída da compilação) está em uma AWS região diferente do projeto de CodeBuild compilação.

Solução recomendada: atualize as configurações do projeto de compilação para apontar para um bucket de saída que esteja na mesma AWS região do projeto de compilação.

Erro: "falha do clone do Git: não é possível acessar '**your-repository-URL**': problema de certificado SSL: certificado autoassinado"

Problema: ao tentar executar um projeto de compilação, a compilação falha com esse erro.

Possível causa: seu repositório de origem tem um certificado autoassinado, mas você não optou por instalar o certificado a partir de seu bucket do S3 como parte do projeto de compilação.

Soluções recomendadas:

- Edite o projeto. Em Certificate, escolha Install certificate from S3. Em Bucket de certificado, selecione o bucket do S3 onde o certificado SSL está armazenado. Em Chave de objeto de certificado, insira o nome da chave do objeto do S3.
- Edite o projeto. Selecione SSL inseguro para ignorar os avisos de SSL ao se conectar ao seu repositório de projetos do GitHub Enterprise Server.

 Note

Recomendamos usar Insecure SSL somente para teste. Ele não deve ser usado em um ambiente de produção.

Erro: "The bucket you are attempting to access must be addressed using the specified endpoint (O bucket que você está tentando acessar deve ser endereçado usando o endpoint especificado)" ao executar uma compilação

Problema: ao executar uma compilação, a fase `DOWNLOAD_SOURCE` de compilação falha com o erro `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

Possível causa: seu código-fonte pré-criado é armazenado em um bucket do S3 e esse bucket está em uma AWS região diferente do projeto de AWS CodeBuild compilação.

Solução recomendada: atualize as configurações do projeto de compilação para apontar para um bucket que contenha o código-fonte preexistente. Certifique-se de que o bucket esteja na mesma AWS região do projeto de construção.

Erro: "This build image requires selecting at least one runtime version" (Esta imagem de compilação requer a seleção de pelo menos um tempo de execução)

Problema: ao executar uma compilação, a fase `DOWNLOAD_SOURCE` de compilação falha com o erro `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

Possível causa: sua compilação usa a versão 1.0 ou posterior da imagem padrão do Amazon Linux 2 (AL2) ou a versão 2.0 ou posterior da imagem padrão do Ubuntu, e um tempo de execução não está especificado no arquivo `buildspec`.

Solução recomendada: se você usar a imagem `aws/codebuild/standard:2.0` CodeBuild gerenciada, deverá especificar uma versão de tempo de execução na `runtime-versions` seção do arquivo `buildspec`. Por exemplo, você pode usar o seguinte arquivo `buildspec` para um projeto que usa PHP:

```
version: 0.2

phases:
  install:
    runtime-versions:
      php: 7.3
  build:
    commands:
      - php --version
artifacts:
  files:
    - README.md
```

Note

Se você especificar uma `runtime-versions` seção e usar uma imagem diferente do Ubuntu Standard Image 2.0 ou posterior, ou da imagem padrão do Amazon Linux 2 (AL2) 1.0 ou posterior, a compilação emitirá o aviso "Skipping install of runtimes. Runtime version selection is not supported by this build image."

Para obter mais informações, consulte [Specify runtime versions in the buildspec file](#).

Erro: "QUEUED: INSUFFICIENT_SUBNET" quando ocorre uma falha em uma compilação em uma fila de compilação

Problema: ocorreu uma falha em uma compilação em uma fila de compilação com um erro semelhante a `QUEUED: INSUFFICIENT_SUBNET`.

Possíveis causas: o bloco IPv4 CIDR especificado para sua VPC usa um endereço IP reservado. Os primeiros quatro endereços IP e o último endereço IP em cada bloco CIDR de sub-rede não estão disponíveis para você usar e não podem ser atribuídos a uma instância. Por exemplo, em uma sub-rede com bloco CIDR 10.0.0.0/24, os seguintes cinco endereços IP são reservados:

- 10.0.0.0: endereço de rede.
- 10.0.0.1: reservado por AWS para o roteador VPC.
- 10.0.0.2: Reservado por AWS. O endereço IP do servidor DNS é sempre a base do intervalo da rede VPC mais dois; no entanto, também reservamos a base de cada intervalo de sub-rede mais dois. Pois VPCs com vários blocos CIDR, o endereço IP do servidor DNS está localizado no CIDR primário. Para ter mais informações, consulte [Amazon DNS server](#) no Manual do usuário da Amazon VPC.
- 10.0.0.3: Reservado AWS por para uso futuro.
- 10.0.0.255: Endereço de transmissão de rede. Não oferecemos suporte à transmissão em uma VPC. Este endereço está reservado.

Soluções recomendadas: verifique se a VPC usa um endereço IP reservado. Substitua o endereço IP reservado por um que não esteja reservado. Para obter mais informações, consulte [Dimensionamento da VPC e da sub-rede](#) no Guia do usuário da Amazon VPC.

Erro: “Não foi possível baixar o cache: RequestError: Falha na solicitação de envio causada por: x509: Falha ao carregar raízes do sistema e nenhuma raiz fornecida”

Problema: ao tentar executar um projeto de compilação, a compilação falha com esse erro.

Possível causa: você configurou o armazenamento em cache como parte do projeto de compilação e está usando uma imagem de docker mais antiga que inclui um certificado raiz expirado.

Solução recomendada: atualize a imagem do Docker que está sendo usada AWS CodeBuild no seu projeto. Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#).

Erro: “Não foi possível baixar o certificado do S3. AccessDenied”

Problema: ao tentar executar um projeto de compilação, a compilação falha com esse erro.

Causas possíveis:

- Você escolheu o bucket do S3 errado para o seu certificado.
- Você digitou a chave de objeto errada para o seu certificado.

Soluções recomendadas:

- Edite o projeto. Em Bucket de certificado, selecione o bucket do S3 onde o certificado SSL está armazenado.
- Edite o projeto. Em Chave de objeto de certificado, insira o nome da chave do objeto do S3.

Erro: "não foi possível localizar as credenciais"

Problema: ao tentar executar AWS CLI, usar um AWS SDK ou chamar outro componente similar como parte de uma compilação, você recebe erros de compilação diretamente relacionados ao AWS CLI AWS SDK ou ao componente. Por exemplo, é possível obter um erro de compilação, como `Unable to locate credentials`.

Causas possíveis:

- A versão do AWS CLI AWS SDK ou componente no ambiente de compilação é incompatível com o AWS CodeBuild
- Você está executando um contêiner do Docker em um ambiente de compilação que usa o Docker, e o contêiner não tem acesso às AWS credenciais por padrão.

Soluções recomendadas:

- Certifique-se de que seu ambiente de compilação tenha a seguinte versão ou superior do AWS CLI AWS SDK ou componente.
 - AWS CLI: 1.10.47
 - AWS SDK para C++: 0.2.19
 - AWS SDK para Go: 1.2.5
 - AWS SDK para Java: 1.11.16
 - AWS SDK para JavaScript: 2.4.7
 - AWS SDK para PHP: 3.18.28

- AWS SDK para Python (Boto3): 1.4.0
 - AWS SDK para Ruby: 2.3.22
 - Botocore: 1.4.37
 - CoreCLR: 3.2.6-beta
 - Node.js: 2.4.7
- Se você precisar executar um contêiner Docker em um ambiente de compilação e o contêiner exigir AWS credenciais, você deverá passar as credenciais do ambiente de compilação para o contêiner. No arquivo buildspec, inclua um comando `run` do Docker como o seguinte. Este exemplo usa o comando `aws s3 ls` para listar seus buckets do S3 disponíveis. A `-e` opção passa pelas variáveis de ambiente necessárias para que seu contêiner acesse AWS as credenciais.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Se você estiver criando uma imagem do Docker e a compilação exigir AWS credenciais (por exemplo, para baixar um arquivo do Amazon S3), você deverá passar as credenciais do ambiente de criação para o processo de criação do Docker da seguinte forma.
1. No Dockerfile do código-fonte da imagem de Docker, especifique as instruções ARG a seguir.

```
ARG AWS_DEFAULT_REGION
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. No arquivo buildspec, inclua um comando `build` do Docker como o seguinte. As `--build-arg` opções definem as variáveis de ambiente necessárias para que seu processo de criação do Docker acesse as AWS credenciais.

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -
t your-image-tag .
```

RequestError erro de tempo limite ao executar CodeBuild em um servidor proxy

Problema: você recebe um erro `RequestError` semelhante a um dos seguintes:

- `RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeout` de CloudWatch Logs.
- `Error uploading artifacts: RequestError: send request failed caused by: Put https://<your-bucket>.s3.<your-aws-region>.amazonaws.com/*: dial tcp 52.219.96.208:443: connect: connection refused` do Amazon S3.

Causas possíveis:

- O `ssl-bump` não está configurado corretamente.
- A política de segurança da sua organização não permite que você use `ssl_bump`.
- Seu arquivo `buildspec` não tem configurações de proxy especificadas usando um elemento `proxy`.

Soluções recomendadas:

- Certifique-se de que o `ssl-bump` está configurado corretamente. Se você usar o Squid para seu servidor de proxy, consulte [Configurar o Squid como um servidor de proxy explícito](#).
- Siga estas etapas para usar endpoints privados para Amazon S3 CloudWatch e Logs:
 1. Na tabela de roteamento da sua sub-rede privada, remova a regra que você adicionou para rotear o tráfego destinado à Internet para o seu servidor de proxy. Para obter informações, consulte [Creating a subnet in your VPC](#) no Guia do usuário da Amazon VPC.
 2. Crie um endpoint privado do Amazon S3 e um endpoint CloudWatch Logs e associe-os à sub-rede privada do seu Amazon VPC. Para obter informações, consulte [VPC endpoint services](#) no Guia do usuário da Amazon VPC.
 3. Confirme que Habilitar nome de DNS privado na Amazon VPC está selecionada. Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário da Amazon VPC.
- Se você não usar o `ssl-bump` para um servidor de proxy explícito, adicione uma configuração de proxy ao arquivo `buildspec` usando um elemento `proxy`. Para obter mais informações, consulte [Execute CodeBuild em um servidor proxy explícito](#) e [Sintaxe de buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
```

```
phases:
  build:
    commands:
```

O bourne shell (sh) deve existir em imagens de compilação

Problema: você está usando uma imagem de compilação que não é fornecida por AWS CodeBuild, e suas compilações falham com a mensagem `Build container found dead before completing the build`.

Possível causa: o shell Bourne (sh) não está incluído na sua imagem de compilação. CodeBuild precisa sh executar comandos e scripts de construção.

Solução recomendada: se sh não estiver presente em sua imagem de compilação, não se esqueça de incluí-la antes de iniciar outras compilações que usem sua imagem. (CodeBuild já inclui sh em suas imagens de construção.)

Aviso: "ignorando a instalação de tempos de execução. A seleção de versão de tempo de execução não é compatível com esta imagem de compilação" ao executar uma compilação

Problema: ao executar uma compilação, o log de compilação contém este aviso.

Possível causa: sua compilação não usa a versão 1.0 ou posterior da imagem padrão do Amazon Linux 2 (AL2) ou a versão 2.0 ou posterior da imagem padrão do Ubuntu, e um tempo de execução é especificado em uma `runtime-versions` seção no seu arquivo `buildspec`.

Solução recomendada: certifique-se de que seu arquivo `buildspec` não contenha uma seção `runtime-versions`. A `runtime-versions` seção só é necessária se você usar a imagem padrão do Amazon Linux 2 (AL2) ou posterior ou a imagem padrão do Ubuntu versão 2.0 ou posterior.

Erro: "Não foi possível verificar a JobWorker identidade" ao abrir o CodeBuild console

Problema: Quando você abre o CodeBuild console, uma mensagem de erro "Não foi possível verificar a JobWorker identidade" é exibida.

Possível causa: o perfil do IAM usado para acesso ao console tem uma tag com `jobId` como chave. Essa chave de tag está reservada CodeBuild e causará esse erro se estiver presente.

Solução recomendada: altere todas as tags personalizadas do perfil do IAM que têm a chave `jobId` para ter uma chave diferente, como `jobIdentifier`.

Falha ao iniciar a compilação

Problema: ao iniciar uma compilação, você recebe uma mensagem de erro de falha ao iniciar a compilação.

Possível causa: o número de compilações simultâneas foi atingido.

Soluções recomendadas: espere até que outras compilações sejam concluídas ou aumente o limite de compilação simultânea para o projeto e inicie a compilação novamente. Para obter mais informações, consulte [Configuração de projetos](#).

Acessando GitHub metadados em compilações armazenadas em cache localmente

Problema: em alguns casos, o diretório `.git` em uma compilação em cache é um arquivo de texto e não um diretório.

Possíveis causas: Quando o cache de origem local está habilitado para uma compilação, CodeBuild cria um gitlink para o diretório. `.git` Isso significa que o diretório é `.git`, na verdade, um arquivo de texto que contém o caminho para o diretório.

Soluções recomendadas: em todos os casos, use o comando a seguir para obter o diretório de metadados do Git. Esse comando funcionará independentemente do formato de `.git`:

```
git rev-parse --git-dir
```

AccessDenied: O proprietário do bucket do grupo de relatórios não corresponde ao proprietário do bucket do S3...

Problema: Ao fazer o upload de dados de teste para um bucket do Amazon S3 CodeBuild, não é possível gravar os dados de teste no bucket.

Causas possíveis:

- A conta especificada para o proprietário do bucket do grupo de relatórios não corresponde ao proprietário do bucket do Amazon S3.
- O perfil de serviço não tem acesso de gravação ao bucket.

Soluções recomendadas:

- Altere o proprietário do bucket do grupo de relatórios para corresponder ao proprietário do bucket do Amazon S3.
- Modifique o perfil de serviço para conceder acesso de gravação ao bucket do Amazon S3.

Erro: “Suas credenciais não têm um ou mais escopos de privilégios necessários” ao criar um projeto com CodeBuild CodeConnections

Problema: ao criar um CodeBuild projeto com CodeConnections, você não tem permissão para instalar um webhook do Bitbucket.

Causas possíveis:

- O novo escopo da permissão pode não ter sido aceito em sua conta do Bitbucket.

Soluções recomendadas:

- Para aceitar a nova permissão, você deve ter recebido qualquer e-mail com o assunto Ação necessária - Escopos de alteração enviado pelo Bitbucket,. AWS CodeStar notifications-noreply@bitbucket.org O e-mail contém um link para conceder permissões ao webhook para sua instalação existente do aplicativo CodeConnections Bitbucket.
- Se você não conseguir localizar o e-mail, conceda a permissão navegando até https://bitbucket.org/site/addons/reauthorize?account=<workspace-name>&addon_key=aws-codestar ou https://bitbucket.org/site/addons/reauthorize?addon_key=aws-codestar selecionando o espaço de trabalho ao qual gostaria de conceder permissão ao webhook.

**AWS CodeStar requests access**

This app is hosted at <https://codestar-connections.webhooks.aws>

Read your account information

Read and modify your repositories and their pull requests

Administer your repositories

Read and modify your repositories' webhooks

Authorize for workspace

Allow AWS CodeStar to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

Grant access Cancel

Erro: “Desculpe, nenhum terminal foi solicitado - não foi possível obter a entrada” ao compilar com o comando de instalação do Ubuntu

Problema: [se você estiver executando compilações privilegiadas de contêiner de GPU, talvez esteja instalando o NVIDIA Container Toolkit seguindo este procedimento](#). Na versão mais recente da CodeBuild imagem, CodeBuild pré-instala e configura o docker nvidia-container-toolkit com a imagem mais recente e selecionada. amazonlinux ubuntu Seguir esse procedimento fará com que as compilações com o comando de instalação do Ubuntu falhem com o seguinte erro:

```
Running command curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | gpg --
dearmor --no-tty -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
gpg: Sorry, no terminal at all requested - can't get input
curl: (23) Failed writing body
```

Possíveis causas: A chave gpg já existe no mesmo local.

Soluções recomendadas: O já `nvidia-container-toolkit` está instalado na imagem. Se você ver esse erro, você pode pular o processo de instalação e reinicialização do docker em seu `buildspec`.

Cotas para AWS CodeBuild

As tabelas a seguir listam as cotas atuais em AWS CodeBuild. Essas cotas são para cada AWS região compatível com cada AWS conta, a menos que especificado de outra forma.

Cotas de serviço

A seguir estão as cotas padrão para o AWS CodeBuild serviço.

Name	Padrão	Ajuste	Descrição
Tags associadas por projeto	Cada região compatível: 50	Não	Número máximo de tags que você pode associar a um projeto de build
Projetos de compilação	Cada região compatível: 5.000	Sim	Número máximo de criação de projetos
Tempo limite de compilação em minutos	Cada região compatível: 2.160	Não	Tempo limite máximo de compilação em minutos
Solicitação simultânea de informações sobre compilações	Cada região compatível: 100	Não	Número máximo de compilações sobre as quais você pode solicitar informações a qualquer momento usando a AWS CLI ou AWS um SDK.
Solicitações simultâneas de informações sobre projetos de construção	Cada região compatível: 100	Não	Número máximo de projetos de compilação sobre os quais você pode solicitar informações a qualquer momento usando a AWS CLI ou um AWS SDK.
Compilações em execução simultânea para o ambiente ARM Lambda/10GB	Cada região compatível: 1	Sim	Número máximo de compilações em

Name	Padrão	Ajuste	Descrição
			execução simultânea para o ambiente ARM Lambda/10GB
Compilações em execução simultânea para o ambiente ARM Lambda/1GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM Lambda/1GB
Compilações em execução simultânea para o ambiente ARM Lambda/2GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM Lambda/2GB
Compilações em execução simultânea para o ambiente ARM Lambda/4GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM Lambda/4GB
Compilações em execução simultânea para o ambiente ARM Lambda/8GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM Lambda/8GB
Compilações em execução simultânea para o ambiente ARM/2 XLarge	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM/2 XLarge

Name	Padrão	Ajuste	Descrição
Compilações em execução simultânea para o ambiente ARM/grande	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM/grande
Compilações em execução simultânea para o ambiente ARM/Medium	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM/Medium
Compilações em execução simultânea para o ambiente ARM Lambda/pequeno	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM/pequeno
Compilações em execução simultânea para ARM/ambiente XLarge	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente ARM/XLarge
Compilações em execução simultânea para o ambiente Linux GPU grande	Cada região compatível: 0	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux GPU/grande
Compilações em execução simultânea para o ambiente Linux GPU pequeno	Cada região compatível: 0	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux GPU/pequeno

Name	Padrão	Ajuste	Descrição
Compilações em execução simultânea para o ambiente Linux Lambda/10GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux Lambda/10GB
Compilações em execução simultânea para o ambiente Linux Lambda/1GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux Lambda/1GB
Compilações em execução simultânea para o ambiente Linux Lambda/2GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux Lambda/2GB
Compilações em execução simultânea para o ambiente Linux Lambda/4GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux Lambda/4GB
Compilações em execução simultânea para o ambiente Linux Lambda/8GB	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux Lambda/8GB
Compilações em execução simultânea para o ambiente Linux/2 XLarge	Cada região compatível: 0	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux/2 XLarge

Name	Padrão	Ajuste	Descrição
Compilações em execução simultânea para o ambiente Linux/grande	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux/grande
Compilações em execução simultânea para o ambiente Linux/médio	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Linux/médio
Compilações em execução simultânea para o ambiente Linux/pequeno	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para ambientes Linux/pequeno
Compilações em execução simultânea para Linux/ambiente XLarge	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para Linux/ambiente XLarge
Compilações em execução simultânea para o ambiente Windows Server 2019/grande	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2019/grande
Compilações em execução simultânea para o ambiente Windows Server 2019/médio	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2019/médio

Name	Padrão	Ajuste	Descrição
Compilações em execução simultânea para o ambiente Windows Server 2022/2 XLarge	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2022/2 XLarge
Compilações em execução simultânea para o ambiente Windows Server 2022/Large	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2022/Large
Compilações em execução simultânea para o ambiente Windows Server 2022/Medium	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2022/Medium
Compilações em execução simultânea para o ambiente Windows Server 2022/XLarge	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows Server 2022/ XLarge
Compilações em execução simultânea para o ambiente Windows/grande	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows/grande
Compilações em execução simultânea para o ambiente Windows/médio	Cada região compatível: 1	Sim	Número máximo de compilações em execução simultânea para o ambiente Windows/médio

Name	Padrão	Ajuste	Descrição
Período mínimo para o tempo limite de compilação em minutos	Cada região compatível: 5	Não	Tempo limite mínimo de compilação em minutos
Grupos de segurança na configuração de VPC	Cada região compatível: 5	Não	Grupos de segurança disponíveis para configuração de VPC
Sub-redes na configuração de VPC	Cada região compatível: 16	Não	Sub-redes disponíveis para configuração de VPC

 Note

As métricas internas determinarão as cotas padrão para compilações em execução simultânea.

As cotas para o número máximo de compilações em execução simultânea variam dependendo do tipo de computação. Para alguns tipos de computação e plataformas, o padrão é 20. Para solicitar uma cota maior de compilações simultâneas ou se receber o erro “Não é possível ter mais de X compilações ativas na conta”, use o link acima para fazer a solicitação. Para obter mais informações sobre a definição de preço, consulte [Preços do AWS CodeBuild](#).

Outros limites

Projetos de compilação

Recurso	Padrão
Caracteres permitidos na descrição de projeto de build	Any
Caracteres permitidos em um nome de projeto de build	As letras A-Z e a-z, os números 0-9 e os caracteres especiais - e _

Recurso	Padrão
Tamanho de um nome de projeto de compilação	2 a 150 caracteres, inclusive
Tamanho máximo de uma descrição de projeto de build	255 caracteres
Número máximo de relatórios que você pode adicionar a um projeto	5
Número de minutos que você pode especificar para o limite de build de todos os builds relacionados em um projeto de build.	5 a 2.160 (36 horas)

Compilações

Recurso	Padrão
Tempo máximo em que o histórico de uma compilação é mantido	1 ano
Número de minutos que você pode especificar para o limite de build em um único build	5 a 2.160 (36 horas)

Frotas de computação

Recurso	Padrão
Número simultâneo de frotas de computação	10
Instâncias em execução simultânea para frotas do ambiente ARM/Small	1
Instâncias em execução simultânea para frotas do ambiente ARM/Large	1

Recurso	Padrão
Instâncias em execução simultânea para frotas do ambiente Linux/Small	1
Instâncias em execução simultânea para frotas do ambiente Linux/Medium	1
Instâncias em execução simultânea para frotas do ambiente Linux/Large	1
Instâncias em execução simultânea para frotas de Linux/ambiente XLarge	1
Instâncias em execução simultânea para frotas de ambientes Linux/2 XLarge	0
Instâncias em execução simultânea para frotas do ambiente GPU/Small	0
Instâncias em execução simultânea para frotas do ambiente GPU/Large	0
Instâncias em execução simultânea para frotas do ambiente Windows Server 2019/Medium	1
Instâncias em execução simultânea para frotas do ambiente Windows Server 2019/Large	1
Instâncias em execução simultânea para frotas do ambiente Windows Server 2022/Medium	1
Instâncias em execução simultânea para frotas do ambiente Windows Server 2022/Large	1
Instâncias em execução simultânea para frotas do ambiente Mac ARM/Medium	1
Instâncias em execução simultânea para frotas do ambiente Mac ARM/Large	1

Relatórios

Recurso	Padrão
Duração máxima da disponibilidade de um relatório de teste após sua criação	30 dias
Tamanho máximo da mensagem de um caso de teste	5.000 caracteres
Tamanho máximo do nome de um caso de teste	Mil caracteres
Número máximo de grupos de relatórios por AWS conta	5.000
Número máximo de casos de teste por relatório	500

Tags

Os limites de tags se aplicam às tags em projetos de CodeBuild criação e recursos de grupos de CodeBuild relatórios.

Recurso	Padrão
Nomes de chave de tags de recurso	Qualquer combinação de letras, números, espaços e caracteres Unicode permitidos em UTF-8, com tamanho entre 1 e 127 caracteres. Os caracteres permitidos são + - = . _ : / @ Os nomes de chaves de tag devem ser exclusivos, e cada chave só pode ter um valor. Um nome de chave de tag não pode: • começar com aws : • consistir apenas de espaços

Recurso	Padrão
	• terminar com um espaço • conter emojis ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Valores da tag de recurso	Qualquer combinação de letras, números, espaços e caracteres Unicode permitidos em UTF-8, com tamanho entre 0 e 255 caracteres. Os caracteres permitidos são + - = . _ : / @ Uma chave pode ter apenas um valor, mas várias chaves podem ter o mesmo valor. Um valor de chave de tag não pode conter emojis ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;

AWS CodeBuild Histórico do documento do Guia do Usuário

A tabela a seguir descreve as mudanças importantes na documentação desde a última versão do AWS CodeBuild. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

- Versão mais recente da API: 06-10-2016

Alteração	Descrição	Data
Conteúdo atualizado: políticas AWS gerenciadas (predefinidas) para AWS CodeBuild	As AWSCode BuildRead OnlyAccess políticas AWSCode BuildAdminAccess AWSCodeBuildDevelo perAccess,, e foram atualizad as. O recurso original arn:aws:codeconnec tions:*:*:connecti on/* foi atualizado paraarn:aws:codeconnec tions:*:*:* .	10 de junho de 2025
Nova referência para chaves de CodeBuild condição	Foi adicionada uma nova página de referência com exemplos de uso de chaves de CodeBuild condição. Veja as chaves de AWS CodeBuild condição .	15 de maio de 2025
Novo conteúdo: Exemplo de servidor de criação de imagens Docker para CodeBuild	CodeBuild agora suporta o descarregamento de suas compilações do Docker para um servidor gerenciado de criação de imagens.	15 de maio de 2025

Novo tipo de computação: CUSTOM_INSTANCE_TYPE	CodeBuild agora permite que você crie uma frota de capacidade reservada com um tipo de instância específico usando <code>CUSTOM_INSTANCE_TYPE</code> .	23 de abril de 2025
Novo suporte para CodeBuild sandbox	Foram adicionadas informações sobre o uso da nova CodeBuild sandbox. Consulte Depurar compilações com CodeBuild sandbox .	7 de abril de 2025
Novos tipos de ambiente Windows	CodeBuild agora oferece suporte aos tipos de ambiente Windows XL e 2XL. Para obter mais informações, consulte Build environment compute types .	31 de março de 2025
Cache atualizado do Amazon S3	CodeBuild agora oferece suporte a um novo comportamento de cache para o armazenamento em cache do Amazon S3.	28 de março de 2025
Novo conteúdo: opções de configuração do GitHub Actions Runner	CodeBuild agora suporta <code>CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME</code> o registro no nível corporativo.	11 de março de 2025
Novo conteúdo: adicionar novo tipo de filtro de webhook	Adicione suporte a um novo tipo de filtro de webhook (<code>ORGANIZATION_NAME</code>).	11 de março de 2025

Novo conteúdo: Tutorial para assinatura de código da Apple com Fastlane com armazenamento de certificados S3	Adicione um novo tutorial para assinatura de código da Apple com Fastlane no CodeBuild uso do S3 para armazenamento de certificados	5 de fevereiro de 2025
Novo conteúdo: Tutorial para assinatura de código da Apple com Fastlane com armazenamento de GitHub certificados	Adicione um novo tutorial para assinatura de código da Apple com Fastlane em CodeBuild uso GitHub para armazenamento de certificados	5 de fevereiro de 2025
Novo conteúdo: Buildkite runner	Adicione novo conteúdo para o corredor Buildkite	31 de janeiro de 2025
Novo conteúdo: webhooks manuais do Buildkite	Adicione suporte para webhooks manuais do Buildkite.	31 de janeiro de 2025
Novo conteúdo: referência de buildspec do Batch build	Adicione suporte para compilações em lote em frotas de capacidade reservada e ambientes Lambda.	8 de janeiro de 2025
Novo conteúdo: execute testes paralelos em compilações em lote	Adicione novo conteúdo para testes paralelos em compilações em lote.	2 de janeiro de 2025
Novo conteúdo: repetir as compilações automaticamente	CodeBuild agora suporta repetição automática para compilações de webhook.	18 de dezembro de 2024
Novo conteúdo: configure uma credencial de registro privada para executores auto-hospedados	Adicione suporte para definir credenciais de registro ao usar imagens personalizadas de registros não privados.	13 de dezembro de 2024

Novo conteúdo: opções de configuração do GitHub Actions Runner	CodeBuild GitHub Os corredores auto-hospedados do Actions agora permitem que você registre seus corredores no nível da organização e configure um ID de grupo de corredores específico.	12 de dezembro de 2024
Novo conteúdo: Adicionar atributo em caso de falha RETRY	CodeBuild agora permite que você configure um atributo de falha RETRY em seu buildspec.	12 de dezembro de 2024
Novo conteúdo: GitLab webhooks manuais	Adicione suporte para GitLab webhooks manuais.	11 de dezembro de 2024
Conteúdo atualizado: aliases atualizados	Atualize aliases para imagens de tempo de execução padrão baseadas em Linux.	22 de novembro de 2024
Conteúdo atualizado: substituições de rótulos são compatíveis com o CodeBuild executor hospedado GitLab	Adicione suporte para substituições personalizadas de rótulos de imagens para GitLab corredores.	22 de novembro de 2024
Conteúdo atualizado: substituições de rótulos suportadas pelo executor de ações CodeBuild hospedado GitHub	Adicione suporte para substituições personalizadas de rótulos de imagens para executores de GitHub ações.	22 de novembro de 2024

Conteúdo atualizado: políticas AWS gerenciadas (predefinidas) para AWS CodeBuild	As AWSCode BuildRead OnlyAccess políticas AWSCode BuildAdminAccess AWSCodeBuildDeveloperAccess,, e foram atualizadas. O recurso original <code>arn:aws:codebuild:*</code> foi atualizado para <code>arn:aws:codebuild:*:*:project/*</code> .	15 de novembro de 2024
Conteúdo atualizado: capacidade reservada	As frotas de capacidade reservada agora oferecem suporte a construções sem contêineres: ARM EC2 EC2, Linux e Windows. EC2	12 de novembro de 2024
Conteúdo atualizado: capacidade reservada	As frotas de capacidade reservada agora oferecem suporte à computação baseada em atributos.	6 de novembro de 2024
Novo conteúdo: repetir as compilações automaticamente	CodeBuild agora permite que você ative a repetição automática de suas compilações.	25 de outubro de 2024
Novo conteúdo: executado CodeBuild em um servidor proxy gerenciado para frotas de capacidade reservada	Adicione suporte a configurações de proxy para frotas de capacidade reservada.	15 de outubro de 2024
Novo conteúdo: corredores autogerenciados GitLab	Adicione novo conteúdo para corredores autogerenciados GitLab	17 de setembro de 2024
Novo conteúdo: webhooks em GitLab grupo	Adicione suporte para webhooks em GitLab grupo.	17 de setembro de 2024

Novo conteúdo: executar os comandos buildspec nas fases INSTALL, PRE_BUILD e POST_BUILD	Adicionar suporte a <code>-with-buildspec</code> .	20 de agosto de 2024
Conteúdo atualizado: capacidade reservada	As frotas com capacidade reservada agora oferecem suporte ao macOS.	19 de agosto de 2024
Novo conteúdo: conexões de GitHub aplicativos	Adicione suporte para conexões de GitHub aplicativos.	14 de agosto de 2024
Novo conteúdo: conexões da aplicação do Bitbucket	Adicione suporte para conexões da aplicação do Bitbucket.	14 de agosto de 2024
Novo conteúdo: Vários tokens de acesso em CodeBuild	Adicione suporte para fornecer tokens de acesso a provedores terceirizados a partir de segredos em AWS Secrets Manager ou por meio de Conexões de código da AWS conexões.	14 de agosto de 2024
Conteúdo atualizado: capacidade reservada	As frotas de capacidade reservada agora oferecem suporte aos tipos de XLarge computação ARM Medium XLarge, ARM e ARM 2.	5 de agosto de 2024
Conteúdo atualizado: capacidade reservada	CodeBuild agora oferece suporte à conectividade VPC para frotas de capacidade reservada no Windows.	1.º de agosto de 2024

Novos tipos de computação ARM	CodeBuild agora oferece suporte aos tipos de XLarge computação ARM Medium XLarge, ARM e ARM 2. Para obter mais informações, consulte Build environment compute types .	10 de julho de 2024
Conteúdo atualizado: assinatura SHA	Atualize a assinatura do Secure Hash Algorithm (SHA) para x86_64 e ARM.	19 de junho de 2024
Novo conteúdo: GitHub webhooks globais e organizacionais	Adicione suporte para GitHub webhooks globais e organizacionais.	17 de junho de 2024
Novo conteúdo: adicionar novo tipo de filtro de webhook	Adicione suporte a um novo tipo de filtro de webhook (REPOSITORY_NAME).	17 de junho de 2024
Espaço em disco atualizado	Os tipos de computação ARM Small e ARM Large agora aumentaram o espaço em disco.	4 de junho de 2024
Novo conteúdo: GitHub webhooks manuais	Adicione suporte para GitHub webhooks manuais.	23 de maio de 2024
Conteúdo atualizado: capacidade reservada	CodeBuild agora oferece suporte à conectividade VPC para frotas de capacidade reservada no Amazon Linux.	15 de maio de 2024
Conteúdo atualizado: Imagens de computação do Lambda	Adicione suporte do Lambda para .NET 8 (a1-lambda/aarch64/dotnet8 e a1-lambda/x86_64/dotnet8)	8 de maio de 2024

Cota atualizada: tempo limite de compilação	Atualize a cota máxima de tempo limite de compilação para 2160 minutos (36 horas).	1º. de maio de 2024
Conteúdo atualizado: políticas AWS gerenciadas (predefinidas) para AWS CodeBuild	As AWSCode BuildRead OnlyAccess políticas AWSCode BuildAdminAccess AWSCodeBuildDevelo perAccess,, e foram atualizadas para refletir o Conexões de código da AWS rebranding.	30 de abril de 2024
Novo conteúdo: Senha da aplicação ou token de acesso do Bitbucket	Adicione suporte a tokens de acesso do Bitbucket.	11 de abril de 2024
Novo conteúdo: Relatórios de descoberta automática em CodeBuild	CodeBuild agora oferece suporte à descoberta automática de relatórios.	4 de abril de 2024
Novo conteúdo: executores de GitHub ações auto-hospedados	Adicione novo conteúdo para executores de GitHub ações auto-hospedados	2 de abril de 2024
Novo conteúdo: GitLab conexões	Adicione suporte GitLab e conexões GitHub autogerenciadas.	25 de março de 2024
Novo conteúdo: adicionar novos eventos de webhook e tipos de filtro	Adicione suporte a novos eventos de webhook (RELEASED e PRERELEASED) e tipos de filtro (TAG_NAME e RELEASE_NAME).	15 de março de 2024
Novo conteúdo: adicionar um novo evento de webhook: PULL_REQUEST_CLOSED	Adicione suporte a um novo evento de webhook: PULL_REQUEST_CLOSED .	20 de fevereiro de 2024

Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild	Adicionar suporte ao Windows Server Core 2019 (windows-base:2019-3.0)	7 de fevereiro de 2024
Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild	Adicionar suporte para novos runtimes para Amazon Linux 2023 (a12/aarch64/standard/3.0)	29 de janeiro de 2024
Novo conteúdo: capacidade reservada	CodeBuild agora suporta frotas de capacidade reservada em CodeBuild.	18 de janeiro de 2024
Novo tipo de computação	CodeBuild agora oferece suporte a um tipo de XLarge computação Linux. Para obter mais informações, consulte Build environment compute types .	8 de janeiro de 2024
Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild	Adicionar suporte para novos runtimes para Amazon Linux 2 (a12/standard/5.0) e Ubuntu (ubuntu/standard/7.0)	14 de dezembro de 2023
Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild	Adicionar suporte para novas imagens de computação do Lambda	8 de dezembro de 2023
Novo conteúdo: AWS Lambda computação	Adicionar novo conteúdo para o AWS Lambda computador	6 de novembro de 2023
Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild	Adicionar suporte para o Amazon Linux 2 (a12/standard/5.0)	17 de maio de 2023

[Alterações nas políticas gerenciadas para CodeBuild](#)

Detalhes sobre as atualizações das políticas AWS gerenciadas já CodeBuild estão disponíveis. Para ter mais informações, consulte [Atualizações da CodeBuild para políticas gerenciadas pela AWS](#).

16 de maio de 2023

[Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild](#)

Remover o suporte para Amazon Linux 2 (al2/standard/3.0) e adicionar suporte para Amazon Linux 2 (al2/standard/corretto8) e Amazon Linux 2 (al2/standard/corretto11)

9 de maio de 2023

[Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild](#)

Adicionar suporte para Ubuntu 22.04 (ubuntu/standard/7.0)

13 de abril de 2023

[Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild](#)

Remover o suporte para Ubuntu 18.04 (ubuntu/standard/4.0) e Amazon Linux 2 (al2/aarch64/standard/1.0)

31 de março de 2023

[Conteúdo atualizado: remover a limitação da VPC](#)

Removendo a seguinte limitação: se você configurar CodeBuild para trabalhar com uma VPC, o armazenamento em cache local não é suportado. A partir de 28/02/22, sua compilação de VPC levará mais tempo, pois uma nova EC2 instância da Amazon será usada para cada compilação.

1 de março de 2023

[Conteúdo atualizado: Imagens do Docker fornecidas por CodeBuild](#)

Remover o suporte para Ubuntu 18.04 (ubuntu/standard/3.0) e Amazon Linux 2 (al2/standard/2.0)

30 de junho de 2022

[Exemplo do Amazon ECR: restringir o acesso à imagem](#)

Quando CodeBuild as credenciais são usadas para extrair uma imagem do Amazon ECR, você pode restringir o acesso à imagem a um projeto específico CodeBuild . Para obter mais informações, consulte [Amazon ECR sample](#).

10 de março de 2022

[Adição de suporte à região](#)

O tipo de computação o ARM_CONTAINER é compatível com as seguintes regiões adicionais: Ásia-Pacífico (Seul), Canadá (Central) , Europa (Londres) e Europa (Paris). Para obter mais informações, consulte [Build environment compute types](#).

10 de março de 2022

[Nova limitação da VPC](#)

Se você configurar CodeBuild para trabalhar com uma VPC, o armazenamento em cache local não será suportado. A partir de 28/02/22, sua compilação de VPC levará mais tempo, pois uma nova EC2 instância da Amazon será usada para cada compilação.

25 de fevereiro de 2022

[Modo de relatório em lote](#)

CodeBuild agora permite que você selecione como os status de compilação em lote são enviados ao provedor de origem de um projeto. Para obter mais informações, consulte [Batch report mode](#).

4 de outubro de 2021

[Novo tipo de computação](#)

CodeBuild agora oferece suporte a um pequeno tipo de computação ARM. Para obter mais informações, consulte [Build environment compute types](#).

13 de setembro de 2021

[Projetos de compilação pública](#)

CodeBuild agora permite que você disponibilize os resultados de compilação de seus projetos de compilação ao público sem precisar acessar uma AWS conta. Para obter mais informações, consulte [Public build projects](#).

11 de agosto de 2021

[Depuração de sessão para compilações em lote](#)

CodeBuild agora oferece suporte à depuração de sessão para compilações em lote. Para obter mais informações, consulte [build-graph](#) e [build-list](#).

3 de março de 2021

[Limite de compilação simultânea em nível de projeto](#)

CodeBuild agora permite limitar o número de compilações simultâneas para um projeto de construção. Para obter mais informações, consulte [Configuração do projeto concurrentBuildLimite](#).

16 de fevereiro de 2021

[Nova propriedade buildspec: s3-prefix](#)

CodeBuild agora fornece a propriedade `s3-prefix` buildspec para artefatos que permite especificar um prefixo de caminho para artefatos que são enviados para o Amazon S3. Para obter mais informações, consulte [s3-prefix](#).

9 de fevereiro de 2021

[Nova propriedade buildspec: on-failure](#)

CodeBuild agora fornece a propriedade `on-failure` buildspec para as fases de construção que permite determinar o que acontece quando uma fase de construção falha. Para obter mais informações, consulte [on-failure](#).

9 de fevereiro de 2021

Nova propriedade buildspec: exclude-paths	CodeBuild agora fornece a propriedade <code>exclude-paths</code> buildspec para artefatos que permite excluir caminhos de seus artefatos de construção. Para obter mais informações, consulte exclude-paths .	9 de fevereiro de 2021
Nova propriedade buildspec: enable-symlinks	CodeBuild agora fornece a propriedade <code>enable-symlinks</code> buildspec para artefatos que permite preservar links simbólicos em um artefato ZIP. Para obter mais informações, consulte enable-symlinks .	9 de fevereiro de 2021
Aprimoramento do nome do artefato de buildspec	CodeBuild agora permite que a <code>artifacts/name</code> propriedade contenha informações do caminho. Para obter mais informações, consulte nome .	9 de fevereiro de 2021
Relatórios de cobertura de código	CodeBuild agora fornece relatórios de cobertura de código. Para obter mais informações, consulte Relatórios de cobertura de código .	30 de julho de 2020

Compilações em lote	CodeBuild agora suporta a execução simultânea e coordenada de compilações de um projeto. Para obter mais informações, consulte Batch builds in CodeBuild .	30 de julho de 2020
Imagem do Windows Server 2019	CodeBuild agora fornece uma imagem de compilação do Windows Server Core 2019. Para obter mais informações, consulte Imagens do Docker fornecidas por CodeBuild .	20 de julho de 2020
Session Manager	CodeBuild agora permite pausar uma compilação em execução e, em seguida, usar o Gerenciador de AWS Systems Manager Sessões para se conectar ao contêiner de compilação e visualizar o estado do contêiner. Para obter mais informações, consulte Session Manager .	20 de julho de 2020
Tópico atualizado	CodeBuild agora suporta a especificação de um shell para usar em seus ambientes de construção no arquivo buildspec. Para obter mais informações, consulte Referência de especificação de compilação .	25 de junho de 2020

Relatórios de teste com frameworks de teste	Foram adicionados vários tópicos que descrevem como gerar relatórios CodeBuild de teste com várias estruturas de teste. Para obter mais informações, consulte Relatórios de teste com estruturas de trabalho de teste .	29 de maio de 2020
Tópicos atualizados	CodeBuild agora suporta a adição de tags aos grupos de relatórios. Para obter mais informações, consulte ReportGroup .	21 de maio de 2020
Suporte a relatórios de teste	CodeBuild o suporte para relatórios de teste agora está disponível ao público em geral.	21 de maio de 2020
Tópicos atualizados	CodeBuild agora suporta a criação de filtros de webhook para Github e Bitbucket que acionam compilações somente quando a mensagem de confirmação principal corresponde à expressão especificada. Para obter mais informações, consulte amostra de GitHub pull request e filtro de webhook e amostra de pull request e filtro de webhook do Bitbucket .	6 de maio de 2020

Novos tópicos

CodeBuild agora oferece suporte ao compartilhamento de recursos de projetos de compilação e grupos de relatórios. Para obter mais informações, consulte [Como trabalhar com projetos compartilhados](#) e [Como trabalhar com grupos de relatórios compartilhados](#).

13 de dezembro de 2019

Tópicos novos e atualizados

CodeBuild agora oferece suporte a relatórios de teste durante a execução de um projeto de compilação. Para obter mais informações, consulte [Trabalhando com relatórios de teste](#), [Criar um relatório de teste](#) e [Criar um relatório de teste usando a AWS CLI amostra](#).

25 de novembro de 2019

Tópico atualizado

CodeBuild agora oferece suporte aos tipos de ambiente Linux GPU e Arm e ao tipo de 2xlarge computação. Para obter mais informações, consulte [Build environment compute types](#).

19 de novembro de 2019

Tópicos atualizados

CodeBuild agora suporta números de compilação em todas as compilações, exportação de variáveis de ambiente e AWS Secrets Manager integração. Para obter mais informações, consulte [Variáveis exportadas](#) e [Secrets Manager](#) em [Sintaxe da buildspec](#).

6 de novembro de 2019

Novo tópico

CodeBuild agora suporta regras de notificação. Você pode usar regras de notificação para notificar os usuários sobre alterações importantes em projetos de compilação. Para obter mais informações, consulte [Criar uma regra de notificação](#).

5 de novembro de 2019

Tópicos atualizados

CodeBuild agora suporta os tempos de execução do Android versão 29 e Go versão 1.13. Para obter mais informações, consulte [Imagens do Docker fornecidas pelo CodeBuild](#) e a [Sintaxe da buildspec](#).

10 de setembro de 2019

Tópicos atualizados

Ao criar um projeto, agora você pode escolher a imagem gerenciada do Amazon Linux 2 (AL2). Para obter mais informações, consulte [Imagens de Docker fornecidas pelo CodeBuild](#) e [Versões de tempo de execução no exemplo de arquivo buildspec para o CodeBuild](#).

16 de agosto de 2019

Tópico atualizado

Quando você cria um projeto, pode optar por desativar a criptografia dos registros do S3 e, se você usar um repositório de origem baseado no Git, inclua os submódulos do Git. Para obter mais informações, consulte [Criar um projeto de compilação em CodeBuild](#).

8 de março de 2019

Novo tópico

CodeBuild agora oferece suporte ao cache local. Você pode especificar o armazenamento em cache local em um ou mais dos quatro modos ao criar uma compilação. Para obter mais informações, consulte [Criar cache em CodeBuild](#).

21 de fevereiro de 2019

Novos tópicos	CodeBuild agora oferece suporte a grupos de filtros de webhook para especificar eventos que acionam uma compilação. Para obter mais informações, consulte Filtrar eventos de GitHub webhook e Filtrar eventos de webhook do Bitbucket .	8 de fevereiro de 2019
Novo tópico	O Guia CodeBuild do usuário agora mostra como usar CodeBuild com um servidor proxy. Para obter mais informações, consulte Usar CodeBuild com um servidor proxy .	4 de fevereiro de 2019
Tópicos atualizados	CodeBuild agora suporta o uso de uma imagem do Amazon ECR que está em outra AWS conta. Vários tópicos foram atualizados para refletir essa mudança, incluindo uma amostra do Amazon ECR para CodeBuild , Criar um projeto de construção e Criar uma função CodeBuild de serviço .	24 de janeiro de 2019

[Suporte a registros privados do Docker](#)

CodeBuild agora suporta o uso de uma imagem do Docker que é armazenada em um registro privado como seu ambiente de execução. Para obter mais informações, consulte [Private registry with AWS Secrets Manager sample](#).

24 de janeiro de 2019

[Tópico atualizado](#)

CodeBuild agora suporta o uso de um token de acesso para se conectar aos repositórios GitHub (com um token de acesso pessoal) e do Bitbucket (com uma senha de aplicativo). Para obter mais informações, consulte [Criar um projeto de compilação \(console\)](#) e [Usar tokens de acesso com seu provedor de origem](#).

6 de dezembro de 2018

[Tópico atualizado](#)

CodeBuild agora oferece suporte a novas métricas de compilação que medem a duração de cada fase em uma compilação. Para obter mais informações, consulte [CodeBuild CloudWatch métricas](#).

15 de novembro de 2018

Tópico da política do endpoint da VPC	Por CodeBuild enquanto, os endpoints do Amazon VPC oferecem suporte às políticas . Para obter mais informações, consulte Criar uma política de VPC endpoint para o CodeBuild .	9 de novembro de 2018
Conteúdo atualizado	Os tópicos foram atualizados para refletir a nova experiência de console.	30 de outubro de 2018
Exemplo do Amazon EFS	CodeBuild pode montar um sistema de arquivos Amazon EFS durante uma compilação usando comandos no arquivo buildspec de um projeto. Para obter mais informações, consulte o exemplo do Amazon EFS para CodeBuild .	26 de outubro de 2018
Webhooks do Bitbucket	CodeBuild agora suporta webhooks quando você usa o Bitbucket para seu repositório. Para obter mais informações, consulte Exemplo de solicitação pull do Bitbucket para o CodeBuild .	2 de outubro de 2018
Logs do S3	CodeBuild agora oferece suporte a registros de compilação em um bucket do S3. Anteriormente, você só podia criar registros usando o CloudWatch Logs. Para obter mais informações, acesse Criar projeto .	17 de setembro de 2018

[Várias origens de entrada e artefatos de saída](#)

CodeBuild agora oferece suporte a projetos que usam mais de uma fonte de entrada e publicam mais de um conjunto de artefatos. Para obter mais informações, consulte [Várias fontes de entrada e amostra de artefatos de entrada](#) e [CodePipeline integração com CodeBuild e várias fontes de entrada e amostra de artefatos de saída](#).

30 de agosto de 2018

[Exemplo de versionamento de semântica](#)

O Guia CodeBuild do usuário agora tem uma amostra baseada em casos de uso que demonstra como usar o controle de versão semântico para criar nomes de artefatos no momento da construção. Para obter mais informações, consulte [Usar versionamento semântico para nomear o exemplo de artefatos de compilação](#).

14 de agosto de 2018

[Novo exemplo de site estático](#)

O Guia CodeBuild do usuário agora tem uma amostra baseada em casos de uso que demonstra como hospedar a saída de compilação em um bucket do S3. O exemplo aproveita o suporte recente dos artefatos de compilação não criptografados. Para obter mais informações, consulte [Criar um site estático com saída de compilação hospedada em um bucket do S3](#).

14 de agosto de 2018

[Suporte para substituir o nome de um artefato pelo versionamento semântico](#)

Agora você pode usar o controle de versão semântico para especificar um formato CodeBuild usado para nomear artefatos de construção. Isso é útil porque um artefato de compilação com um nome codificado substitui os artefatos de compilação anteriores que usam o mesmo nome codificado. Por exemplo, se uma compilação for acionada várias vezes por dia, agora você poderá adicionar um carimbo de data e hora ao nome do artefato. Cada nome de artefato de compilação é exclusivo e não substitui os artefatos de compilações anteriores.

7 de agosto de 2018

[Suporte de artefatos de compilação não criptografados](#)

CodeBuild agora oferece suporte a compilações com artefatos de compilação não criptografados. Para obter mais informações, consulte [Create a build project \(console\)](#).

26 de julho de 2018

[Support para CloudWatch métricas e alarmes da Amazon](#)

CodeBuild agora fornece integração com CloudWatch métricas e alarmes. Você pode usar o CloudWatch console CodeBuild ou para monitorar compilações no nível do projeto e da conta. Para obter mais informações, consulte [Monitoramento de compilações](#).

19 de julho de 2018

[Suporte para informar o status de uma compilação](#)

CodeBuild agora pode relatar o status do início e da conclusão de uma compilação ao seu provedor de origem. Para obter mais informações, consulte [Criar um projeto de compilação em CodeBuild](#).

10 de julho de 2018

[Variáveis de ambiente adicionadas à CodeBuild documentação](#)

A página [Variáveis de ambiente em ambientes de compilação](#) foi atualizada com as variáveis de ambiente CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH e CODEBUILD_START_TIME.

9 de julho de 2018

[Suporte para um bloco finally no arquivo buildspec](#)

A CodeBuild documentação foi atualizada com detalhes sobre o finally bloco opcional em um arquivo buildspec . Comandos no bloco final são sempre executados após os comandos no bloco de comandos correspondente. Para obter mais informações, consulte [Sintaxe da buildspec](#).

20 de junho de 2018

[CodeBuild notificações de atualização do agente](#)

A CodeBuild documentação foi atualizada com detalhes sobre como você pode usar o Amazon SNS para ser notificado quando novas versões do CodeBuild agente forem lançadas. Para obter mais informações, consulte [Receber notificações para novas versões do AWS CodeBuild agente](#).

15 de junho de 2018

Atualizações anteriores

A tabela a seguir descreve alterações importantes em cada versão do Guia do usuário do AWS CodeBuild antes de junho de 2018.

Alteração	Descrição	Data
Suporte para compilações do Windows	CodeBuild agora oferece suporte a compilações para a plataforma Microsoft Windows Server, incluindo um ambiente de compilação pré-empacotado para o.NET	25 de maio de 2018

Alteração	Descrição	Data
	Core 2.0 no Windows. Para obter mais informações, consulte Execute amostras do Microsoft Windows para CodeBuild .	
Suporte para idempotência da compilação	Ao executar o <code>start-build</code> comando com o AWS Command Line Interface (AWS CLI), você pode especificar que a compilação é idempotente. Para obter mais informações, consulte Executar uma compilação (AWS CLI) .	15 de maio de 2018
Suporte para a substituição de mais configurações de projetos de compilação	Agora você pode substituir mais configurações de projetos de compilação ao criar uma compilação. As substituições são apenas para essa compilação. Para obter mais informações, consulte Execute AWS CodeBuild compilações manualmente .	15 de maio de 2018
Suporte para endpoint da VPC	Agora você pode usar VPC endpoints para aumentar a segurança das compilações. Para obter mais informações, consulte Usar VPC endpoints .	18 de março de 2018

Alteração	Descrição	Data
Suporte para acionadores	Agora você pode criar acionadores para programar compilações com frequências normais. Para obter mais informações, consulte Crie AWS CodeBuild gatilhos .	28 de março de 2018
Documentação dos endpoints do FIPS	Agora você pode aprender como usar o AWS Command Line Interface (AWS CLI) ou um AWS SDK para CodeBuild instruir o uso de um dos quatro endpoints do Federal Information Processing Standards (FIPS). Para obter mais informações, consulte Especifique o AWS CodeBuild endpoint .	28 de março de 2018
AWS CodeBuild disponível na Ásia-Pacífico (Mumbai), Europa (Paris) e América do Sul (São Paulo)	AWS CodeBuild agora está disponível nas regiões Ásia-Pacífico (Mumbai), Europa (Paris) e América do Sul (São Paulo). Para ter mais informações, consulte AWS CodeBuild no Referência geral da Amazon Web Services.	28 de março de 2018

Alteração	Descrição	Data
GitHub Suporte ao Enterprise Server	CodeBuild agora pode criar a partir do código-fonte armazenado em um repositório do GitHub Enterprise Server. Para obter mais informações, consulte Execute a amostra do GitHub Enterprise Server .	25 de janeiro de 2018
Suporte à profundidade de clone Git	CodeBuild agora suporta a criação de um clone superficial com um histórico truncado para o número especificado de confirmações. Para obter mais informações, consulte Criar um projeto de compilação .	25 de janeiro de 2018
Suporte à VPC	As compilações habilitadas para VPC já podem acessar recursos dentro da VPC. Para obter mais informações, consulte Suporte à VPC .	27 de novembro de 2017
Suporte ao armazenamento em cache de dependências	CodeBuild agora suporta o cache de dependências. Isso permite CodeBuild salvar certas partes reutilizáveis do ambiente de compilação no cache e usá-las em todas as compilações.	27 de novembro de 2017

Alteração	Descrição	Data
Suporte aos emblemas de compilação	CodeBuild agora suporta o uso de emblemas de construção, que fornecem uma imagem incorporável e gerada dinamicamente (emblema) que exibe o status da versão mais recente de um projeto. Para obter mais informações, consulte Exemplo de emblemas de compilação .	27 de novembro de 2017
AWS Config integração	AWS Config agora oferece suporte CodeBuild como um AWS recurso, o que significa que o serviço pode rastrear seus CodeBuild projetos. Para obter mais informações sobre AWS Config, consulte AWS Config amostra .	20 de outubro de 2017
Reconstrua automaticamente o código-fonte atualizado nos repositórios GitHub	Se seu código-fonte estiver armazenado em um GitHub repositório, você poderá habilitar AWS CodeBuild a reconstrução do código-fonte sempre que uma alteração de código for enviada para o repositório. Para obter mais informações, consulte Execute a solicitação GitHub pull e a amostra do filtro de webhook .	21 de setembro de 2017

Alteração	Descrição	Data
Novas formas de armazenar e recuperar variáveis de ambiente sensíveis ou grandes no Amazon EC2 Systems Manager Parameter Store	Agora você pode usar o AWS CodeBuild console ou o AWS CLI para recuperar variáveis de ambiente grandes ou confidenciais armazenadas no Amazon EC2 Systems Manager Parameter Store. Agora você também pode usar o AWS CodeBuild console para armazenar esses tipos de variáveis de ambiente no Amazon EC2 Systems Manager Parameter Store. Anteriormente, você poderia recuperar apenas esses tipos de variáveis de ambiente, incluindo-os em uma buildspec ou executando comandos de compilação para automatizar a AWS CLI. Você só pode armazenar esses tipos de variáveis de ambiente usando o console Amazon EC2 Systems Manager Parameter Store. Para obter mais informações, consulte Criar um projeto de compilação, Alterar as configurações do projeto de compilação e Executar compilações manualmente.	14 de setembro de 2017

Alteração	Descrição	Data
Suporte à exclusão de compilação	Você já pode excluir compilações no AWS CodeBuild. Para obter mais informações, consulte Excluir compilações .	31 de agosto de 2017
Forma atualizada de recuperar variáveis de ambiente sensíveis ou grandes armazenadas no Amazon EC2 Systems Manager Parameter Store usando um buildspec	AWS CodeBuild agora fica mais fácil usar um buildspec para recuperar variáveis de ambiente sensíveis ou grandes armazenadas no Amazon EC2 Systems Manager Parameter Store. Anteriormente, você poderia recuperar apenas esses tipos de variáveis de ambiente executando comandos de compilação para automatizar a AWS CLI. Para obter mais informações, consulte o mapeamento parameter-store em Sintaxe de buildspec .	10 de agosto de 2017
AWS CodeBuild suporta Bitbucket	CodeBuild agora é possível criar a partir do código-fonte armazenado em um repositório do Bitbucket. Para obter mais informações, consulte Criar um projeto de compilação e Executar compilações manualmente .	10 de agosto de 2017

Alteração	Descrição	Data
AWS CodeBuild disponível no Oeste dos EUA (Norte da Califórnia), Europa (Londres) e Canadá (Central)	AWS CodeBuild agora está disponível nas regiões Oeste dos EUA (Norte da Califórnia), Europa (Londres) e Canadá (Central). Para ter mais informações, consulte AWS CodeBuild no Referência geral da Amazon Web Services.	29 de junho de 2017
Nomes de arquivo de buildspec e locais alternativos compatíveis	Você já pode especificar um nome de arquivo ou local alternativo de um arquivo de buildspec a ser usado em um projeto de compilação, em vez de um arquivo de buildspec padrão chamado <code>buildspec.yml</code> na raiz do código-fonte. Para obter mais informações, consulte Nome do arquivo buildspec e local de armazenamento .	27 de junho de 2017
Exemplo de notificações de compilação atualizado	CodeBuild agora fornece suporte integrado para notificações de criação por meio do Amazon CloudWatch Events e do Amazon Simple Notification Service (Amazon SNS). O Exemplo de notificações de compilação anterior foi atualizado para demonstrar esse novo comportamento.	22 de junho de 2017

Alteração	Descrição	Data
Docker no exemplo de imagem personalizada adicionado	Uma amostra mostrando como usar CodeBuild e personalizar uma imagem de compilação do Docker para criar e executar uma imagem do Docker foi adicionada. Para mais informações, consulte o Docker no exemplo de imagem personalizada .	7 de junho de 2017
Busque o código-fonte para GitHub pull requests	Quando você executa uma compilação com CodeBuild base no código-fonte armazenado em um GitHub repositório, agora você pode especificar um ID de GitHub pull request para criar. Você também pode especificar um ID de confirmação, um nome de ramificação ou um nome de tag. Para obter mais informações, consulte o valor Versão de origem em Executar uma compilação (console) ou o valor sourceVersion em Executar uma compilação (AWS CLI) .	6 de junho de 2017

Alteração	Descrição	Data
Versão de especificação da compilação atualizada	Uma nova versão do formato de buildspec foi lançada. A versão 0.2 resolve o problema de CodeBuild executar cada comando de compilação em uma instância separada do shell padrão. Além disso, na versão 0.2, <code>environment_variables</code> é renomeado para <code>env</code> , e <code>plaintext</code> é renomeado para <code>variables</code> . Para obter mais informações, consulte Referência de especificação de construção para CodeBuild .	9 de maio de 2017
Dockerfiles para criar imagens disponíveis em GitHub	As definições de muitas das imagens de compilação fornecidas pelo AWS CodeBuild estão disponíveis como Dockerfiles em GitHub. Para obter mais informações, consulte a coluna Definição da tabela em Imagens do Docker fornecidas por CodeBuild .	2 de maio de 2017

Alteração	Descrição	Data
AWS CodeBuild disponível na Europa (Frankfurt), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney) e Ásia-Pacífico (Tóquio)	AWS CodeBuild agora está disponível nas regiões Europa (Frankfurt), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney) e Ásia-Pacífico (Tóquio). Para ter mais informações, consulte AWS CodeBuild no Referência geral da Amazon Web Services.	21 de março de 2017
CodePipeline suporte de ação de teste para CodeBuild	Agora você pode adicionar a um pipeline em CodePipeline uma ação de teste que usa CodeBuild. Para obter mais informações, consulte Adicionar uma ação CodeBuild de teste a um pipeline (CodePipeline console) .	8 de março de 2017
Os arquivos de buildspec oferecem suporte à busca pela saída da compilação em diretórios de nível superior selecionados	Os arquivos Buildspec agora permitem que você especifique diretórios individuais de nível superior cujo conteúdo você pode instruir CodeBuild a incluir nos artefatos de saída da compilação. Isso é feito através do mapeamento <code>base-directory</code> . Para obter mais informações, consulte Sintaxe de buildspec .	8 de fevereiro de 2017

Alteração	Descrição	Data
Variáveis de ambiente integradas	AWS CodeBuild fornece variáveis de ambiente incorporadas adicionais para suas compilações usarem. Entre elas, há variáveis de ambiente que descrevem a entidade que iniciou o build, a URL do repositório de código-fonte, a ID de versão do código-fonte, além de outras variáveis. Para obter mais informações, consulte Variáveis de ambiente em ambientes de compilação .	30 de janeiro de 2017
AWS CodeBuild disponível no Leste dos EUA (Ohio)	AWS CodeBuild agora está disponível na região Leste dos EUA (Ohio). Para ter mais informações, consulte AWS CodeBuild no Referência geral da Amazon Web Services.	19 de janeiro de 2017

Alteração	Descrição	Data
Informações de comportamentos de shell e comando	CodeBuild executa cada comando especificado em uma instância separada do shell padrão de um ambiente de compilação. Esse comportamento padrão pode produzir alguns efeitos colaterais inesperados em seus comandos. Recomendamos algumas abordagens para trabalhar com esse comportamento padrão, se necessário. Para obter mais informações, consulte Shells e comandos em ambientes de compilação .	9 de dezembro de 2016
Informações de variáveis de ambiente	CodeBuild fornece várias variáveis de ambiente que você pode usar em seus comandos de construção. Você também pode definir suas próprias variáveis de ambiente. Para obter mais informações, consulte Variáveis de ambiente em ambientes de compilação .	7 de dezembro de 2016
Tópico da solução de problemas	As informações de solução de problemas agora estão disponíveis. Para obter mais informações, consulte Solução de problemas AWS CodeBuild .	5 de dezembro de 2016

Alteração	Descrição	Data
Lançamento inicial do plugin Jenkins	Esta é a versão inicial do plugin CodeBuild Jenkins. Para obter mais informações, consulte Use AWS CodeBuild com Jenkins .	5 de dezembro de 2016
Lançamento inicial do Guia do usuário	Esta é a versão inicial do Guia do usuário do CodeBuild .	1° de dezembro de 2016

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.